NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**HOW TO OPTIMALLY INTERDICT
A BELLIGERENT PROJECT
TO DEVELOP A NUCLEAR WEAPON**

by

Eric M. Skroch

March 2004

Thesis Advisor: Gerald G. Brown
Second Reader: Robert C. Harney

Approved for public release; distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2004	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: How to Optimally Interdict a Belligerent Project to Develop a Nuclear Weapon	**5. FUNDING NUMBERS**
6. AUTHOR(S) Eric M. Skroch	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

Despite decades of energetic international control efforts, nuclear weapons technology continues to spread world-wide. To understand how these complex weapons programs can be developed, we assume the role of a nation seeking to build a first fission weapon, and the ability to continue to build more. We introduce a large-scale project management model that includes alternate development paths to achieve certain key technical milestones. We show how such a project can be optimally accelerated by expediting critical tasks. Next, we present a new analysis tool to detect vulnerabilities in such a development program: we seek optimal actions to impede, set back and/or otherwise frustrate completion of a first weapon, even if the proliferator knows what we are doing to delay things. This two-sided project evaluation tool is implemented with a combination of off-the-shelf project management software, optimization software and custom code. An illustrative case study of a first fission weapon program shows how this new analysis tool can be used. Our methods also apply to chemical, biological and/or radiological dispersion weapons, as well as to more conventional strategic industrial and commercial activities.

14. SUBJECT TERMS Project management, network interdiction, nuclear nonproliferation	**15. NUMBER OF PAGES** 67
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

THIS PAGE INTENTIONALLY LEFT BLANK

HOW TO OPTIMALLY INTERDICT
A BELLIGERENT PROJECT
TO DEVELOP A NUCLEAR WEAPON

Eric Matthew Skroch
Lieutenant, Supply Corps, United States Navy
B.S., Northwestern University, 1993

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 2004

Author: Eric M. Skroch

Approved by: Gerald G. Brown
Thesis Advisor

Robert C. Harney
Second Reader

James N. Eagle
Chairman, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Despite decades of energetic international control efforts, nuclear weapons technology continues to spread worldwide. To understand how these complex weapons programs can be developed, we assume the role of a nation seeking to build a first fission weapon, and the ability to continue to build more. We introduce a large-scale project management model that includes alternate development paths to achieve certain key technical milestones. We show how such a project can be optimally accelerated by expediting critical tasks. Next, we present a new analysis tool to detect vulnerabilities in such a development program: we seek optimal actions to impede, set back and/or otherwise frustrate completion of a first weapon, even if the proliferator knows what we are doing to delay things. This two-sided project evaluation tool is implemented with a combination of off-the-shelf project management software, optimization software and custom code. An illustrative case study of a first fission weapon program shows how this new analysis tool can be used. Our methods also apply to chemical, biological and/or radiological dispersion weapons, as well as to more conventional strategic industrial and commercial activities.

THIS PAGE INTENTIONALLY LEFT BLANK

THESIS DISCLAIMER

The reader is cautioned that the computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

We introduce a model for planning, even expediting, development of a nation's first operational nuclear weapon. Our goal is to understand how such a complex project can be conducted by a belligerent state so that we might find vulnerabilities to exploit: we seek actions to delay, impede, interdict and/or otherwise frustrate completion of such a weapon of mass destruction.

Preventing the proliferation of such devices is an international priority. The 1968 Treaty on the Non-proliferation of Nuclear Weapons (NPT) has been ratified by 188 nations, more than any other international arms control agreement. Global proliferation concerns have been renewed by recent world events such as North Korea's withdrawal from NPT and revelations of Libya's clandestine work on nuclear weapons.

Nuclear weapon development attracts intense global scrutiny. The International Atomic Energy Agency (IAEA), chartered by the United Nations, is responsible for verifying that nuclear materials are used only for peaceful purposes by enforcing safeguards imposed by the treaty. Any nation that has ratified NPT and is now attempting to develop its first nuclear weapon would therefore almost certainly run a covert program. Outside discovery of such a program would depend on intelligence gathering or a public announcement by the proliferators intended to intimidate other nations into favorable political or economic concessions. When IAEA inspections are denied or a nation like North Korea withdraws from the treaty, the international community may take other steps to counter the proliferation threat.

We introduce a project management tool to identify the tasks in a weapons development program that are most critical to its timely completion. We highlight the effect of delaying an individual task, or set of tasks, on completion of a weapon. These delays might be inflicted by deliberate interdiction of the weapon project; in fact, *we might choose to set back certain milestones already achieved*, as Israel did in 1981 when they successfully bombed Iraq's Osirak reactor facility. In addition to military action, delay-

ing actions might also include sabotage of facilities or supplies, embargo or blockade of material shipments, or motivating key personnel to leave the program.

Mathematics-based methods of scheduling and managing complex projects have been widely employed in government and industry since the late 1950's. The original Program Evaluation Review Technique/Critical Path Method (PERT/CPM) models have been extended in a variety of ways over the years to incorporate situations that arise in different types of projects. Here, we must combine several of these individual embellishments.

The weapon developer, referred to henceforth as the "proliferator," wants to minimize the duration of the project to develop a nuclear weapon by reducing the duration of as many tasks as he can afford. We present a novel and flexible model that captures many of the issues involved in expediting such a sophisticated research and development project. We include the ability to choose among alternatives that are sometimes available for achieving certain milestones, scheduling restrictions based on the availability of scarce resources, and "crashing" the project (i.e., shortening project duration by applying additional resources to selected tasks), among others. Using this tool as a forecast of proliferation plans, we next seek an optimal set of tasks in the project to interdict in order to inflict the most delay in its completion.

The project interdictor wants to impede progress to make the project take as long as possible, subject to constraints on what he can afford to do. He may exert interdiction effort against a particular task in a variety of ways, constrained by a monetary budget or the political, environmental or economic implications of his actions.

We develop a two-sided optimization: the proliferator and the interdictor can each see what the opponent is doing, and can each behave optimally with resources at hand to, respectively, minimize and maximize the time to completion of a first fission weapon. This optimization provides a powerful analytical tool. It can be used to identify the most vulnerable tasks in the project, to suggest alternate delay plans with equivalent delaying effects to a policy maker and to explore the sensitivity of the project duration to the inputs.

Our prototypic decision support system is implemented using off-the-shelf software. All user interaction and data management are done through software already familiar to millions of personal computer users. Powerful optimization software, unfamiliar to many potential users of this product, is hidden behind a straightforward user interface incorporated within standard project management software.

Computational experience with our case study shows that good solutions can be found quickly. "What-if" questions regarding particular delay plans can be answered in a fraction of a second using a desktop personal computer. An analyst can take advantage of this speed to explore a wide range of assumptions and find the tasks that are most critical to the expeditious completion of the proliferator's project.

This low-cost tool can be used *today*. It provides *quantitative* information that can be used to guide policy makers working to stop the proliferation of nuclear, chemical, biological and/or radiological dispersion weapons, or to disrupt more conventional strategic industrial and commercial activities.

In our case study, an uninterdicted first weapon is completed in 260 weeks (5 years). Interdiction of this proliferation program, if kept secret from the proliferator, delays completion to 340 weeks ($6\frac{1}{2}$ years). Finally, if we assume that the proliferator can see what the interdictor is doing, and adjust his plan accordingly, we still delay completion to 324.4 weeks ($6\frac{1}{4}$ years). This demonstrates the value to the interdictor of keeping his actions secret, but, more importantly, exposes the inherent vulnerability of such a large project to a few well-placed interdiction actions. It also demonstrates to the proliferator that a covert project can be completed nearly 15 months faster than one which is known by the enemy and therefore subject to interdiction.

We are suggesting how to organize and manage intelligence data in a cohesive fashion that offers visualization, data base functionality and interoperability with Microsoft Office. The optimization is new, but hidden cleanly from view. An analyst, or even a policy maker, can pose questions, get answers and gain deep insights *without any mathematical background at all*.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I am forever indebted to Professor Jerry Brown for his patient guidance and tireless encouragement. Special thanks are due Professor Bob Harney for his expert assistance with the case study. Essential insight provided by Professors Kevin Wood and Matt Carlyle is greatly appreciated. I would have been unable to achieve this important academic milestone without the unending love and understanding of my wife Deborah and our daughter Emily and for that, I am eternally grateful.

THIS PAGE INTENTIONALLY LEFT BLANK

I. NUCLEAR PROLIFERATION

A. BELLIGERENT STATES SEEK NUCLEAR WEAPONS

We introduce a model for planning, even expediting, development of a nuclear weapon. Our goal is to understand how such a complex project can be conducted by a belligerent state so that we might find vulnerabilities to exploit: we seek actions to delay, impede, interdict and/or otherwise frustrate completion of such a weapon of mass destruction.

Nearly sixty years after the first nuclear weapon was detonated, preventing the proliferation of such devices is an international priority. One hundred eighty-eight nations have ratified the 1968 Treaty on the Non-proliferation of Nuclear Weapons (NPT), more than any other international arms control agreement [United Nations, 2004]. North Korea's recent withdrawal from NPT and Libya's revelations of work on nuclear weapons have renewed global proliferation concerns.

Production of nuclear weapons attracts intense global scrutiny; because of NPT safeguards currently in place, any nation attempting to develop its first nuclear weapon would almost certainly run a covert program. Outside discovery of such a program would depend on intelligence gathering or a public announcement by the proliferators intended to intimidate other nations into favorable political or economic concessions. The International Atomic Energy Agency (IAEA), chartered by the United Nations, is responsible for verifying that nuclear materials are used only for peaceful purposes [United Nations, 1968, Article III]. When IAEA inspections are denied or a nation like North Korea withdraws from the treaty, the international community may take other steps: the successful attack on Iraq's Osirak nuclear facility by the Israeli Air Force in 1981 [e.g., Federation of American Scientists, 2000] is an example. Alarmed by intelligence that Iraq could assemble a functional nuclear weapon within two years, the Israelis chose an immediate military option rather than a diplomatic solution.

B. A PROPOSED COURSE OF ACTION

We introduce a project management tool to identify the tasks in a weapons development program that are most critical to its timely completion. We highlight the effect of delaying an individual task, or set of tasks, on completion of a weapon. These delays might be inflicted by deliberate interdiction of the weapon project: in fact, *we might choose to set back certain milestones already achieved*. Delaying actions might include sabotage of facilities or supplies, embargo or blockade of material shipments, or even military acts. Although our example is nuclear weapon development, our methods apply just as well to chemical, biological and/or radiological dispersion weapons, as well as to more conventional strategic industrial and commercial activities.

Since the late 1950's, mathematics-based methods of scheduling and managing complex projects have been widely employed in government and industry. The original Program Evaluation Review Technique/Critical Path Method (PERT/CPM) models [Malcolm, *et al*, 1959] have been extended over the years to incorporate a variety of situations that arise in different types of projects, such as alternate choices that are sometimes available for achieving certain milestones, scheduling restrictions based on the availability of scarce resources, and "crashing" the project (i.e., shortening project duration by applying additional effort to selected tasks), among other additions. Due to the complexity of nuclear weapons programs, *we need all these embellishments*.

Project management models are universally represented as networks [e.g., Moder, Phillips and Davis, 1983, Chapter 1]. In such a representation, the overall duration of the project is the additive length of the longest path through the network, also known as the *critical path*. The weapon developer, referred to henceforth as the "proliferator," wants to minimize the duration of the project to develop a nuclear weapon by reducing the duration of as many "critical" tasks as he can afford. While building the project, he may be able to choose among several alternate means available to reach some milestone.

The project interdictor wants to impede progress to make the project take as long as possible, subject to constraints on what the interdictor can afford to do. He may exert interdiction effort against a particular task in a variety of ways, constrained by a monetary budget or the political, environmental or economic implications of his actions.

We develop a two-sided optimization: the proliferator and the interdictor can each see what their opponent is doing, and can each behave optimally with resources at hand to, respectively, minimize and maximize the completion of a first fission weapon, and the ability to continue producing more.

THIS PAGE INTENTIONALLY LEFT BLANK

II. MODELING PROLIFERATION AND ITS INTERDICTION

A. PROJECT MANAGEMENT IN NUCLEAR PROLIFERATION

Embarking on a program to develop a nation’s first nuclear weapon requires a significant investment of materials, manpower and technology. Even for a modern industrialized nation, this investment will likely represent a significant fraction of that nation’s available resources. Such a program is a massive undertaking and we assume that it would be subject to intense high-level management scrutiny and careful, centralized coordination.

Moder, Phillips and Davis [1983, p. 3] define a *project* as “a set of tasks or activities related to the achievement of some planned objective, normally where the objective is unique or non-repetitive.” Our proliferator’s program to develop his first nuclear weapon fits this definition well. We can reasonably expect him to employ standard project management tools to plan and organize the project in order to efficiently schedule its tasks. We assume that once the proliferator commits substantial national resources towards the development of a nuclear weapon, he wants to complete his project and obtain the first operational weapon as quickly as possible.

Our proliferator’s problem can be represented most simply as a classic project network enhanced to allow crashing [e.g., Moder, Phillips and Davis, 1983, Chapter 8]. However, we recognize that this model is inadequate to represent many situations that can occur in practical real-world projects. To this classic project network, we add the following embellishments:

- Completion of any task in a “normal” amount of time consumes a fixed amount of one or more non-renewable resources.
- The duration of an individual task may be shortened (“crashed”) by allocation of additional quantities of the required resources. We assume a linear relationship between the amounts of additional resources provided and the duration of the task—more resources accelerates progress. Each task requires some minimum amount of time, no matter how many additional resources are allocated to it.
- Crashing is limited by the availability of each resource and by an overall monetary budget.

- Certain milestone tasks may be achieved via alternate courses of action. When one alternate path is chosen, the tasks in the other alternative(s) need not be completed. A *decision node* is a task in the project network where alternate courses of action diverge.
- The project being delayed may already be partially completed. If a task that was previously in progress or completed is interdicted, *it and all of its successors must be restarted and completed again in order to finish the project*.
- Standard finish-to-start precedence relationships between pairs of tasks are generalized to include start-to-start, finish-to-finish or start-to-finish. There is a finite time lag (or lead) associated with each precedence constraint.
- The project may include a hierarchical grouping of the tasks. A *summary task* represents the tasks grouped below it in a hierarchy (its "children") but does not consume resources or otherwise contribute to the completion of the project. Summary tasks can be given their own precedence relationship(s) with other tasks in the project. Accommodating them requires introducing dummy precedence arcs into the network to ensure a summary task starts no later than any of its children and to preclude finishing until all of its children finish. The summary task must be completed if any of its children are completed and the children must be completed if the summary task is forced to start by other precedence constraints.
- Interdicting a task delays its completion by a constant amount.
- Possible delay plans are limited by a maximum total number of interdictions and by an interdiction "budget." This budget may represent total financial cost or level of geopolitical impact for each interdiction action.

The proliferator's objective is to minimize the length of the critical path in this network. We first determine what the proliferator would do without any threat of interdiction.

B. EXPEDITING A PROJECT WITH NO INTERDICTION

We are now going to formalize these concepts with a generalization of a well-known project management tool.

1. Sets and Index Use

We use an activity on node (AON) formulation of the project network, employing the following sets of tasks, task relationships and resources. In the AON formulation, the nodes in the network represent the individual tasks in the project and the arcs represent the pairwise partial orders between respective predecessor and successor tasks.

$i, j \in N$	tasks (nodes)
$\sigma, \tau \in N$	start and finish milestone tasks, respectively
$DN \subset N$	decision nodes

$ST \subset N$	hierarchical summary tasks
$AN \subseteq N$	all tasks except the decision nodes and summary tasks; $AN = N - DN - ST$
$(i,j) \in A$	precedence relationships: task *i* precedes task *j* (arcs)
$FS \subseteq A$	$(i,j) \in FS$ implies task *i* must finish before task *j* can start
$FF \subseteq A$	$(i,j) \in FF$ implies task *i* must finish before task *j* can finish
$SF \subseteq A$	$(i,j) \in SF$ implies task *i* must start before task *j* can finish
$SS \subseteq A$	$(i,j) \in SS$ implies task *i* must start before task *j* can start
	FS, *FF*, *SF* and *SS* are mutually exclusive and exhaustive subsets of *A*
$CHILD \subset A$	$(i,j) \in CHILD$ if task *j* is subordinate to summary task *i*
$r \in R$	non-renewable resources

2. Data [units]

t	elapsed time since the beginning of the project [days]
p_i	fraction of work on task $i \in N$ that is complete at time *t* [$0 \le p_i \le 1$]
lag_{ij}	time lag for a precedence constraint $(i,j) \in A$ [days]
$\overline{d}_i$, $\underline{d}_i$	duration of task $i \in N$ with no crashing effort and maximal crashing effort, respectively [days]
$\underline{b}_i$, $\overline{b}_i$	minimum and maximum number, respectively, of successor tasks allowed to be chosen at decision node $i \in DN$ [non-negative integer]
$\underline{rr}_{ir}$	minimum resource requirement for resource $r \in R$ to complete task $i \in N$ with no crashing effort [resource *r*-units]
rer_{ir}	resource expediting rate of resource $r \in R$ for task $i \in N$ [resource *r*-units/day]
c_r	unit cost of resource $r \in R$ [\$/resource *r*-unit]
a_r	total availability of resource $r \in R$ [resource *r*-units]
budget	crashing cost budget [\$]
$delay_i$	amount task $i \in N - DN - ST - \sigma - \tau$ is delayed when interdicted [days]
m	a "large" number used to relax the precedence constraints for tasks that are not actually completed; $m = \max_{i \in N}\left(\overline{d}_i + delay_i\right) + \max_{(i,j)\in A}\left(\left|lag_{ij}\right|\right)$ [days]

3. Variables [units]

S_i	earliest start time of task $i \in N$ [days]
D_i	the decision plan; $D_i = 1$ if task $i \in N$ is completed, 0 otherwise [binary]
δ_i	the crashing plan; δ_i is the amount task $i \in N - DN - \sigma - \tau$ is accelerated by crashing [days]

4. Expediting a Project

$$\min_{S,D,\delta} S_\tau - S_\sigma \tag{a0}$$

$$s.t. \quad S_j + m(1-D_i) \geq S_i + (\bar{d}_i - \delta_i) + lag_{ij} \qquad \forall (i,j) \in FS \tag{a1}$$

$$S_j + (\bar{d}_j - \delta_j) + m(1-D_i) \geq S_i + (\bar{d}_i - \delta_i) + lag_{ij} \qquad \forall (i,j) \in FF \tag{a2}$$

$$S_j + (\bar{d}_j - \delta_j) + m(1-D_i) \geq S_i + lag_{ij} \qquad \forall (i,j) \in SF \tag{a3}$$

$$S_j + m(1-D_i) \geq S_i + lag_{ij} \qquad \forall (i,j) \in SS \tag{a4}$$

$$\sum_r \left[c_r \sum_{i \in N-\sigma-\tau} \left(\underline{rr}_{ir} D_i + rer_{ir} \delta_i \right) \right] \leq budget \tag{a5}$$

$$\sum_{i \in N-\sigma-\tau} \left(\underline{rr}_{ir} D_i + rer_{ir} \delta_i \right) \leq a_r \qquad \forall r \in R \tag{a6}$$

$$\underline{b}_i D_i \leq \sum_{j|(i,j) \in A-(CHILD \cap FF)} D_j \leq \bar{b}_i D_i \qquad \forall i \in DN \tag{a7}$$

$$D_j \geq D_i \qquad \forall (i,j) \in A - CHILD \mid i \in (N - DN) \tag{a8}$$

$$D_j \geq D_i \qquad \forall (i,j) \in CHILD \mid \left(\begin{array}{l} (\exists (n,i) \in A \mid ((n,i) \notin CHILD \,\&\, (i,n) \notin CHILD)) \\ \&(not\ \exists (n,j) \in A \mid n \in DN) \end{array} \right) \tag{a9}$$

$$0 \leq \delta_i \leq \bar{d}_i - \underline{d}_i \qquad \forall i \in N - DN - \sigma - \tau \tag{a10}$$

$$D_i \in \{0,1\} \qquad \forall i \in N \tag{a11}$$

$$S_\sigma = 0,\ D_\sigma = 1 \tag{a12}$$

$$D_i = 1 \qquad \forall i \in N \mid p_i > 0 \tag{a13}$$

$$S_i \geq t \qquad \forall i \in N - ST \mid p_i = 0 \tag{a14}$$

$$S_i = t - p_i(\bar{d}_i - \delta_i) \qquad \forall i \in N \mid 0 < p_i < 1 \tag{a15}$$

$$S_i \leq t - (\bar{d}_i - \delta_i) \qquad \forall i \in N \mid p_i = 1 \tag{a16}$$

The proliferator seeks to minimize the overall length of the project, expressed in (a0) as the difference between the finish time and the start time. All of the choices made among alternatives at the decision nodes are captured in the resulting *decision plan*. The *crashing plan* specifies the level of additional effort applied to each task to speed it up. The decision plan and the crashing plan together encompass all of the decisions that the proliferator must make to expedite the project. They are collectively referred to as the *expediting plan*.

Constraints (a1) to (a4) enforce pairwise task partial orders. Each of these constraints features the length of a task, $(\bar{d}_i - \delta_i)$, its normal duration minus the amount of crashing effort. The term $m(1-D_i)$ [Crowston and Thompson, 1967] relaxes its constraint if task *i* is not one of the tasks actually completed in the final decision plan (i.e., a decision takes an alternate course of action).

Constraints (a5) and (a6) respectively enforce the monetary budget and resource availabilities. Constraints (a7) limit the number of alternatives allowed at each decision point. Constraints (a8) ensure that a task is forced to be completed if any of its predecessors are, except for tasks that are children of a summary task. Constraints (a9) force children of a summary task to be completed only if the summary task has predecessors and the child is not directly affected by a decision. (a10) show that each task has a maximum crash effort (that gives it a shortest duration). Task completion decisions are required to be binary in (a11). Constraint (a12) completes the starting task no matter what decisions are made later on and starts the project at time zero.

Constraints (a13) force the decision plan to include every task that has already commenced, reflecting any decisions already made. Constraints (a14) require tasks with no recorded progress to start no sooner than the current time. (a15) set the start time of tasks that have started but not finished while (a16) ensure completed tasks are assigned start times early enough to have finished by the current time.

5. Model Derivation

Each task in the project has a normal duration, the amount of time required to complete without any special expediting effort. Completing a task requires the expenditure of some fixed amount of one or more non-renewable resources (raw materials, energy, labor, machinery, etc.). A fixed amount of each resource is allocated to the entire project. The duration of a task can be shortened down to some non-negative minimum level by applying additional resources to "crash" the task. The proliferator may expend his resources at any time and any rate he chooses, but the cumulative usage of each resource may never exceed the total quantity available. If the project is interdicted and a task is delayed, a fixed amount of time is added to the duration of that task. The relationship between these times is shown in Figure 1.

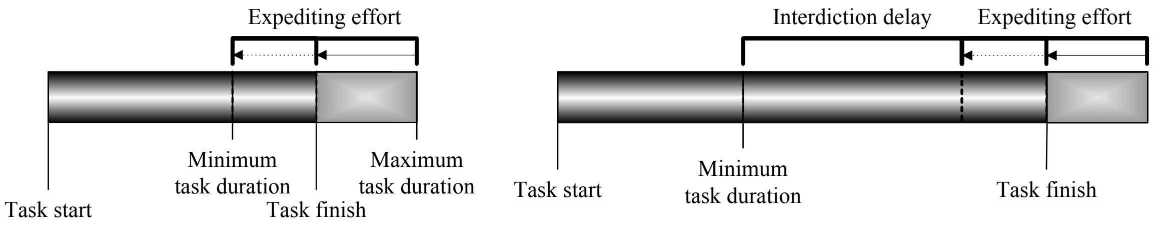


Figure 1. Task duration with expediting effort, both without and including interdiction delay.
The duration of an uninterdicted task (*left*) may be expedited from its longest duration to its shortest. Interdicting the task adds a fixed length of time to its duration (*right*) while still allowing expediting effort to reduce its overall duration.

Early analytical project management work [e.g., Malcolm, *et al*, 1959] assumed that if a task depended on another in any way, the later task could not begin until immediately after all of its prerequisites had been completed. This was later generalized in a model called precedence diagramming by J. David Craig [International Business Machines, 1968]. Precedence diagramming permits every combination of pairwise partial orders between predecessor and successor task start and finish. Given this generality, it is easy to accommodate a lag- or lead-time between pairs of tasks. These relationships are shown in Figure 2.

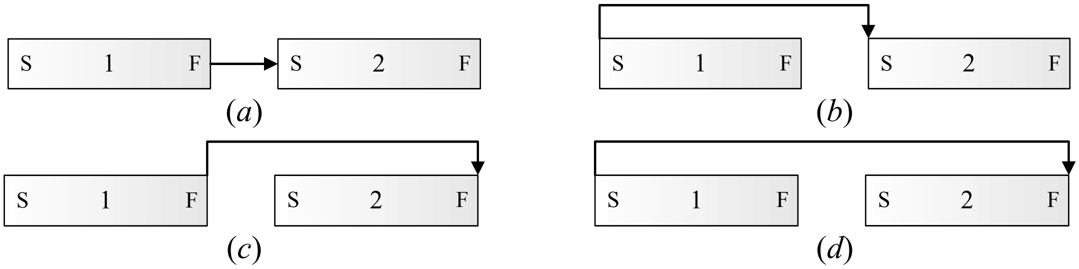


Figure 2. Partial orders between predecessor and successor tasks.
Possible combinations are finish-to-start (*a*), start-to-start (*b*), finish-to-finish (*c*) and start-to-finish (*d*). All lags are shown as positive values (Task 2 "lags" Task 1), although zero and negative lags ("leads") are also allowed.

Crowston and Thompson [1967] introduce decision CPM that allows alternate courses of action to achieve milestones. Their formulation divides the project tasks into sets based on where the decision points are in the project network. Each set of tasks is assigned a binary variable that indicates whether that set of tasks is actually completed in the project. For our purposes, it is more practical to assign each task in the project its own binary decision variable and add additional constraints to maintain the decision set

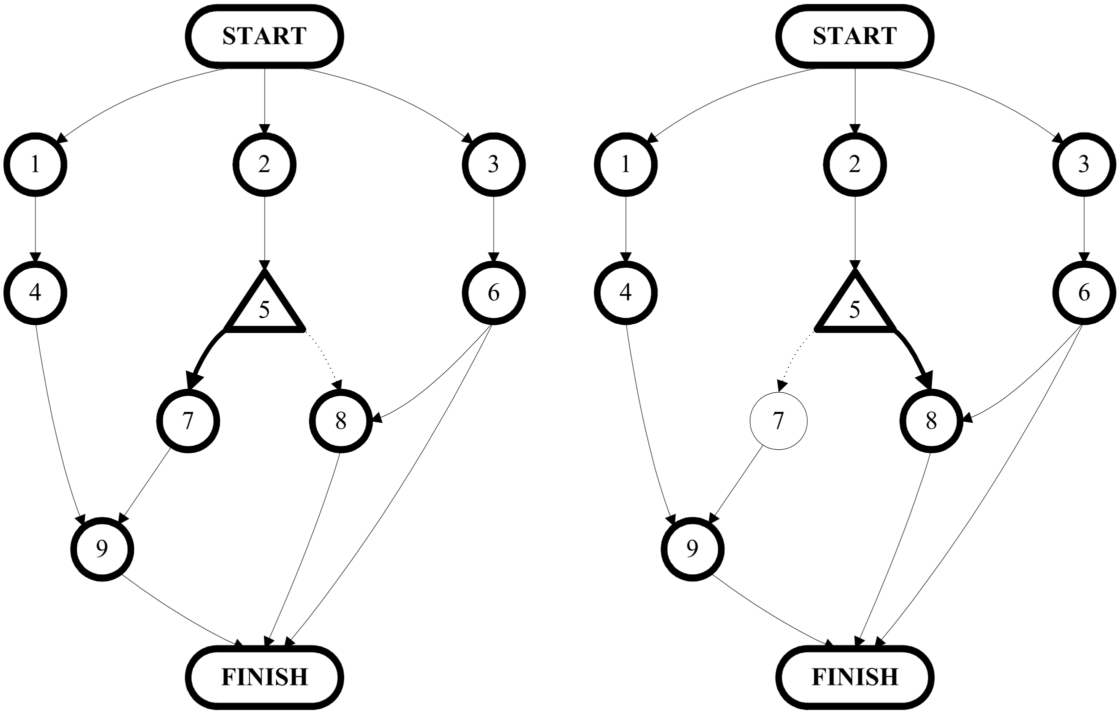


Figure 3. A decision critical path method project network depicting one decision. Similar to the notation introduced by Crowston and Thompson [1967], decision nodes are indicated by triangle. The nodes with the heavy border must be completed in order to finish the project. If Task 7 is chosen (*left*), all tasks are completed while if Task 8 is chosen (*right*), Task 7 is never completed. Dotted lines indicate discretionary precedence constraints that are relaxed.

relationships between them. The effect of a decision on the completion of a project is shown in Figure 3.

C. DELAYING A PROJECT WITH A FIXED DECISION PLAN

The interdictor chooses a set of project tasks to delay in order to maximize the length of the critical path. This model assumes that the proliferator is unaware of the interdiction effort and does nothing to compensate even after the interdictor inflicts a delay. This requires as input the result of expediting the project with no interdiction. *Slack* is calculated for each task from an earliest start time and a latest start time as in conventional CPM. Expediting the project provides the earliest start times directly and latest start times are easily calculated using the length of each expedited task. Tasks on the critical path have zero slack while a positive slack implies the task is not on the critical path.

1. Data [units]

$dCost_i$	cost to delay task $i \in AN - \sigma - \tau$ [\$]
$dBudget$	delaying cost budget [\$]
$maxInt$	maximum number of interdictions allowed [count]
$slack_i$	the difference between the latest start time and earliest start time for task $i \in N$ [days].
$\hat{d}_i$	the fixed decision plan; $\hat{d}_i = 1$ if task $i \in N$ is actually completed, 0 otherwise [binary]

2. Variables [units]

X_i	the delay plan; $X_i = 1$ if task $i \in AN - \sigma - \tau$ is interdicted, 0 otherwise [binary]

3. Delaying a Project

$$\max_X \sum_{i|\hat{d}_i=1} (delay_i - slack_i) X_i \qquad \text{(b0)}$$

$$\sum_{i \in AN-\sigma-\tau} X_i \le maxInt \qquad \text{(b1)}$$

$$\sum_{i \in AN-\sigma-\tau} dCost_i X_i \le dBudget \qquad \text{(b2)}$$

$$X_i \in \{0,1\} \qquad \forall i \in AN - \sigma - \tau \qquad \text{(b3)}$$

The objective function (b0) greedily favors delay plans that lie on the critical path ($\hat{d}_i = 1$ and $slack_i = 0$). The delay added to the project is either the total delay from that task if it is on the expedited critical path or an amount reduced by the slack at that task. Constraint (b1) limits the maximum number of interdictions allowed and (b2) enforces the delaying budget. Constraints (b3) require the interdiction decisions to be binary.

Even though the interdictor may achieve complete surprise with his delaying effort, as Israel did with its attack on Iraq's Osirak reactor facility, the proliferator will almost certainly become aware of the interdiction after the fact. We assume an intelligent adversary will reevaluate his options at this point and take immediate action to mitigate the effects of the interdiction.

D. A TWO-SIDED MODEL: DELAYING A PROJECT WHEN THE PROLIFERATOR AND THE INTERDICTOR CAN SEE EACH OTHER'S ACTIONS

1. Finding the Best Delay Plan

We now assume complete transparency between proliferator and interdictor—each side is fully aware of the other's capabilities, limitations and intentions. We seek a delay plan that maximizes the length of the project even if the proliferator anticipates our

1) Start with proliferator’s optimal plan without interdiction
2) Find a unique interdiction plan
3) Determine the proliferator’s response to this plan
4) Repeat steps 2) through 4) as long as desired or until all possible delay plans have been evaluated
5) Report best solution discovered

Figure 4. Enumeration procedure.

actions and responds optimally. We implement decomposition methods to accomplish this—we enumerate delay plans and then expedite the interdicted project. We refer to the enumeration of delay plans as the *master problem* and to expediting an interdicted project as the *subproblem*.

Figure 4 outlines our enumeration procedure. We start by determining the optimal starting times for each task in the project when there is no interdiction effort whatsoever. The alternatives chosen, task start times, total slack for each task, resource usage levels, the crashing plan employed and the length of the project are recorded, establishing an incumbent baseline to use later. Information about the task slack identifies the critical path and enables the master problem to quickly find candidate delay plans that will have a large impact on the project length. Infeasibility in the subproblem at this stage indicates a problem with the data—insufficient budget or resources, conflict between the elapsed time and progress of certain tasks, or *et cetera*. If the subproblem is feasible, we proceed by initializing a counter that tracks the number of times we solve the master problem.

Our master problem finds a delay plan with the maximum effect on the project length that is different in at least one detail from all the delay plans that have already been considered. For a task that is on the critical path, delaying it adds the full interdiction delay to the length of the project. If a task is not on the critical path, an interdiction has to lengthen it enough to overcome its slack duration before adding anything to the length of the project. Both of these statements assume that the proliferator does not then choose a different course of action that excludes the task in question. Infeasibility in the master problem indicates that we have enumerated every feasible delay plan.

The subproblem evaluates the latest candidate delay plan generated by the master problem. If this delay plan interdicts all of the tasks actually completed in the expedited project, or if every interdicted task is on the resulting critical path, then we have found a lower bound on the amount of delay that the interdictor can inflict. We compute the bound by recomputing the master problem objective using the new slack values from the subproblem. We require all subsequent candidate delay plans to cause at least this much delay to the project. If the subproblem is infeasible, the current delay plan is ideal—*the proliferator has exhausted all of his resources and can never finish the project*. We record the length of the delayed project (infinity if it was infeasible) for this delay plan. Expecting that there may be alternate delay plans that lead to an identical or improved expedited project length, we continue to solve the master problem to find another candidate delay plan.

Each cycle of solving the master problem followed by the subproblem is termed an *iteration*. This cycle repeats until all interdiction options have been explored and the optimal delay plan can be determined, the master problem objective falls below the identified lower bound, or the number of iterations reaches some preset limit. Because we save information about the candidate results of each iteration, it is easy to find and report the best candidates.

2. Data [units]

budgetUsed crashing cost budget consumed prior to time *t* [$]

iter current iteration counter [count]

maxIter maximum number of iterations allowed [count]

totalDelay total delay to project tasks caused by a delay plan [days]

$length_k$ the overall expedited project length in iteration $k \in \{0,\ldots,maxIter\}$ [days]

$slack_{ik}$ the difference between the latest start time and earliest start time for task $i \in N$ in iteration $k \in \{0,\ldots,maxIter\}$ [days].

$\hat{a}_r$ amount of resource $r \in R$ consumed prior to time *t* [resource *r*-units]

$\hat{d}_{ik}$ the decision plan for iteration $k \in \{0,\ldots,maxIter\}$; $\hat{d}_{ik} = 1$ if task $i \in N$ is actually completed in iteration *k*, 0 otherwise [binary]

$\hat{s}_{ik}$ earliest start time of task $i \in N$ in iteration $k \in \{0,\ldots,maxIter\}$ [days]

$\hat{u}_i$ $\hat{u}_i = 1$ if task $i \in N - \sigma$ is a successor of any task in the delay plan being evaluated, 0 otherwise [binary]

$\hat{\delta}_{ik}$ the crashing plan for iteration $k \in \{0,\ldots,maxIter\}$; $\hat{\delta}_{ik}$ is the amount of time saved on task *i* in iteration *k* by expediting its completion [days]

$\hat{x}_{ik}$ the delay plan for iteration $k \in \{0,\ldots,maxIter\}$; $\hat{x}_{ik} = 1$ if task $i \in AN - \sigma - \tau$ is interdicted in iteration *k*, 0 otherwise [binary]

3. Variables [units]

U_i $U_i = 1$ if task $i \in N - \sigma$ is in the current delay plan or is a successor of any task in the delay plan, 0 otherwise [binary]

4. Enumerating Delay Plans

The interdictor chooses a set of project tasks to delay to maximize the length of the longest path in the project network. This model finds a set of tasks to delay.

$$\max_{X} \quad \sum_{i|\hat{d}_{i,iter-1}=1} \left(delay_i - slack_{i,iter-1}\right) X_i + \sum_{i|\hat{d}_{i,iter-1}=0} delay_i X_i \tag{e0}$$

$$s.t. \quad \sum_{i|\hat{x}_{ik}=1} (1 - X_i) + \sum_{i|\hat{x}_{ik}=0} X_i \geq 1 \qquad \forall k \in \{0,\ldots,iter-1\} \tag{e1}$$

$$\sum_{i \in AN-\sigma-\tau} X_i \leq maxInt \tag{e2}$$

$$\sum_{i \in AN-\sigma-\tau} dCost_i X_i \leq dBudget \tag{e3}$$

$$X_i \in \{0,1\} \qquad \forall i \in AN - \sigma - \tau \tag{e4}$$

Given the newest expediting plan, the objective function (e0) evaluates candidate delay plans. If a task is completed in the newest expedited plan, then the new delay added is either the total delay from that task if it was on the expedited critical path or an amount reduced by the slack at that task. Tasks not included in the newest expediting plan are given a maximal delay because, in some situations, a combination of other interdictions could force the proliferator to choose a different decision plan that includes tasks that were previously excluded.

There is a diversity constraint (e1) for each legacy delay plan that requires the Hamming distance [Hamming, 1986] between each legacy delay plan and any new delay plan to be at least one (i.e., a new delay plan must differ in at least one detail). Master problems with diversity constraints are not new [e.g., see the "covering decomposition" of Israeli and Wood, 2002]. Constraints (e2) through (e4) are identical to constraints (b1) through (b3).

Our enumeration *algorithm* will eventually evaluate every feasible delay plan in the search for an optimal solution. We stop the enumeration if we discover a solution we can prove is optimal, or acceptably close to optimal (within some moderating *decomposition tolerance*). However, this algorithm has exponential worst-case runtime. If we place an *a priori* limit on the number of enumeration steps, we have a non-cyclic heuristic search with domain limited by interdiction resource availability and diversity among all candidate delay plans. Our master problem objective, which essentially guides the search, expresses a myopic, greedy goal: maximally delay the most-recently expedited plan. Other objectives may provide faster termination.

5. Finding Tasks Affected by a Delay Plan

Because we assume the interdictor's actions to delay a task will force the proliferator to restart any tasks affected by the delay plan, we identify those tasks before proceeding. A solution to this model associates a "mark" with each affected task:

$$\begin{array}{lllr}
\min\limits_{U} & \sum\limits_{i} U_i & & \text{(c0)} \\
s.t. & U_i = 1 & \forall i \in N \mid \hat{x}_{i,iter} = 1 & \text{(c1)} \\
 & U_j \geq U_i & \forall (i,j) \in A - (CHILD \cap SS) & \text{(c2)} \\
 & U_i \in \{0,1\} & \forall i \in N - \sigma & \text{(c3)}
\end{array}$$

The objective (c0) seeks to minimize the number of tasks marked. Constraints (c1) mark all of the tasks in the current delay plan while each constraint (c2) forces a task to be marked if any of its predecessors are. Constraints (c3) require the marks to be binary. A preferable alternative to this overly formal integer linear programming formulation is to find the successor tasks using a depth-first search algorithm [e.g., Ahuja, *et al*, 1993, pp. 73-76]. The search algorithm marks an interdicted task and then follows the path of its successors until reaching the finish task, marking each unmarked task visited along the way and then repeating this process for each interdicted task in the incumbent delay plan.

6. Expediting an Interdicted Project

Once we have determined the affected tasks, we determine the optimal expediting plan using updated task data.

$$\min_{S,D,\delta}\ S_\tau - S_\sigma \tag{d0}$$

$$s.t.\quad S_j + m(1-D_i) \geq S_i + \left(\left(1-p_i(1-\hat{u}_i)\right)\left(\bar{d}_i - \delta_i\right) + delay_i\hat{x}_{iK}\right) + lag_{ij} \qquad \forall(i,j)\in FS \tag{d1}$$

$$S_j + \left(\left(1-p_j\left(1-\hat{u}_j\right)\right)\left(\bar{d}_j - \delta_j\right) + delay_j\hat{x}_{jK}\right) + m(1-D_i) \geq S_i + \left(\left(1-p_i(1-\hat{u}_i)\right)\left(\bar{d}_i - \delta_i\right) + delay_i\hat{x}_{iK}\right) + lag_{ij} \qquad \forall(i,j)\in FF \tag{d2}$$

$$S_j + \left(\left(1-p_j\left(1-\hat{u}_j\right)\right)\left(\bar{d}_j - \delta_j\right) + delay_j\hat{x}_{jK}\right) + m(1-D_i) \geq S_i + lag_{ij} \qquad \forall(i,j)\in SF \tag{d3}$$

$$S_j + m(1-D_i) \geq S_i + lag_{ij} \qquad \forall(i,j)\in SS \tag{d4}$$

$$\sum_r \left[c_r \sum_{i\in N-\sigma-\tau} \left(1-p_i\left(1-\hat{u}_i\right)\right)\left(\underline{rr}_{ir}D_i + rer_{ir}\delta_i\right)\right] \leq budget - budgetUsed \tag{d5}$$

$$\sum_{i\in N-\sigma-\tau}\left(1-p_i(1-\hat{u}_i)\right)\left(\underline{rr}_{ir}D_i + rer_{ir}\delta_i\right) \leq a_r - \hat{a}_r \qquad \forall r\in R \tag{d6}$$

$$\underline{b}_i D_i \leq \sum_{j|(i,j)\in A-(CHILD\cap FF)} D_j \leq \bar{b}_i D_i \qquad \forall i\in DN \tag{d7}$$

$$D_j \geq D_i \qquad \forall(i,j)\in A-CHILD \mid i\in(N-DN) \tag{d8}$$

$$D_j \geq D_i \qquad \forall(i,j)\in CHILD \mid \left(\begin{array}{l} \left(\exists(n,i)\in A \mid \left((n,i)\notin CHILD \,\&\, (i,n)\notin CHILD\right)\right) \\ \&\left(not\ \exists(n,j)\in A \mid n\in DN\right) \end{array}\right) \tag{d9}$$

$$0 \leq \delta_i \leq \bar{d}_i - \underline{d}_i \qquad \forall i\in N-DN-\sigma-\tau \tag{d10}$$

$$D_i \in \{0,1\} \qquad \forall i\in N \tag{d11}$$

$$S_\sigma = 0,\ D_\sigma = 1 \tag{d12}$$

$$S_i = \hat{s}_i \qquad \forall i\in N \mid (\hat{s}_i < t \text{ and } \hat{u}_i = 0) \tag{d13}$$

$$\delta_i = \hat{\delta}_i \qquad \forall i\in N \mid (\hat{s}_i < t \text{ and } \hat{u}_i = 0) \tag{d14}$$

$$S_i \geq t \qquad \forall i\in N-ST \mid \hat{u}_i = 1 \tag{d15}$$

As in the case with no delaying effort, the proliferator seeks to minimize the overall length of the project, expressed in (d0). Constraints (d1) through (d6) are similar to constraints (a1) through (a6) from the interdiction-free expediting model, but have been modified to account for the effects of a delay plan on the cost and duration of each task. The term $\left(1-p_i(1-\hat{u}_i)\right)$ restores the full cost and duration of any task affected by the delay plan, and accounts for the remaining cost and duration (the total minus any cost expended or progress achieved before time *t*) of any task not affected by the delay plan. The budget and resource constraints, (d5) and (d6), deduct the amounts consumed up to the current time, computed with the assumption of no interdiction delays, from their respective budgets. Constraints (d7) through (d12) are identical to constraints (a7) through (a12).

Constraints (a13) through (a16) have been removed and replaced by (d13) through (d15) that force the resulting crashing plan to respect the progress that was made before the development project was discovered and the delay plan formed. Constraints (d13) and (d14) fix the start times and crashing effort of all tasks that were in progress or complete before the interdiction and are not affected by the delay plan. Constraints (d15) require that all tasks affected by the delay plan start no earlier than the current time. This effectively restarts work on any tasks that were in progress before, but were interrupted by the interdiction. The proliferator is not restricted from choosing a different decision plan and abandoning (partially) completed tasks, if required.

III. A COVERT PROJECT TO DEVELOP A FISSION BOMB

A. A CASE STUDY

A rogue nation is suspected of pursuing the development of a uranium fission weapon. The United States National Command Authority is keenly interested in delaying this weapons program for as long as possible. The range of delaying tactics is considerable, from diplomatic pressure, to asking the United Nations to impose sanctions, to covert or even overt military strikes.

The basics of nuclear weapon design are now well known and publicly available: many of the details from the early weapons programs in the United States and elsewhere have been declassified and published in the open literature. This reduces the basic physics research required to develop a bomb, but unless it is possible to buy weapons-grade uranium or plutonium on the open market (not very likely given the tight international controls in place today), developing an organic capability to produce these materials requires a substantial investment in industrial infrastructure.

B. ASSUMPTIONS

We postulate a medium-sized Southwest Asian country with a primarily agrarian economy [Harney, 2003]. The population is generally well educated by several modern, well-equipped research universities. Most of the nation's industrial capacity is concentrated in a half-dozen large cities. Many industries are underdeveloped by Western standards, but are growing steadily in size and capability. In addition to extensive oil and gas production, the chemical industries are modern and well developed. The country has substantial reserves of uranium ore and is a well-established producer of both ore and concentrated uranium "yellowcake" for the international market. The country has ratified the Nuclear Nonproliferation Treaty and all IAEA safeguards are in place.

Currently, 50% of their electrical power generation is from oil- or gas-fired plants, 25% from hydroelectric dams and 25% from nuclear power. Power demands are growing steadily due to population and industrial growth. Because the growth in power demand is geographically distant from the oil and gas production centers and hydroelectric production is already maximized, the most practical way to satisfy the growing demand for electricity is to expand nuclear generation capacity. The nation already has three operational commercial nuclear power plants with two more under construction. They also have one working research reactor affiliated with a university.

The military is not large, but has sufficient capability to defend against small regional threats. The army has a number of short- and medium-range conventionally armed missiles. The air force has no heavy bombers, but is equipped with a few dozen fighters and medium-range fighter-bombers. The navy has a number of small coastal defense craft. The nation possesses a modest conventional munitions production capability.

This country seeks its own nuclear weapon to counter growing threats from its neighbors. At least three other countries in the region are known or suspected to possess a nuclear arsenal and one nuclear-armed neighbor is growing more hostile. There is also a strong desire to be able to use the nuclear weapons as a bargaining tool to obtain acquiescence and cooperation from several smaller neighbors and to gain credibility on the international stage as a key player in the region. The consensus among military and political advisors is that a total nuclear stockpile of several dozen weapons will be sufficient to achieve these goals.

Because nuclear power generation is vital to the country's economy and grows more so each year, they have no desire to immediately jeopardize this by openly violating the NPT. Thus, any weapons program must be covert and completely independent of any of the existing nuclear facilities that are being monitored by the IAEA. History shows that most nations seeking nuclear weapons have followed this route and established separate military programs rather than divert nuclear material from their safeguarded civilian facilities [NERAC, 2001].

Because uranium mining and yellowcake production are not subject to NPT safeguards [U.S. Congress OTA, 1993, p. 137], the safest path to a weapons program is through development of a uranium enrichment capability fed by existing yellowcake production. As illustrated in Figure 5, this will provide the raw fissile material required for either of two types of uranium fission weapon (gun or implosion) as well as being able to produce fuel for the commercial reactors once their NPT violations have been discovered and United Nations sanctions are imposed. Given the proliferator's goal of completing

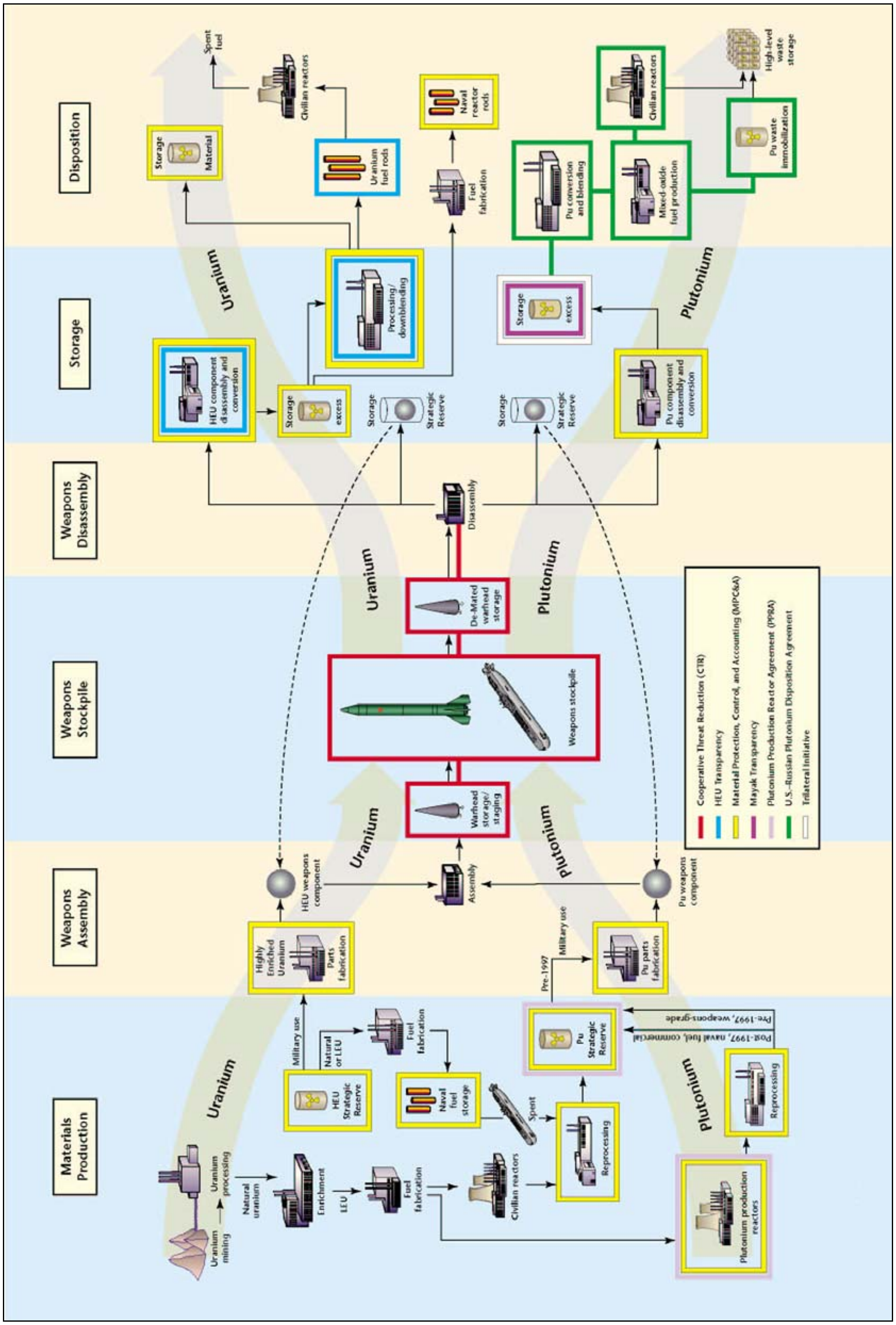


Figure 5. Developing a nuclear weapon [from Spears, 2001].
We study this up through the weapons stockpile.

his first weapon as soon as possible, we assume that he will first pursue a gun-type weapon, the same design used by the "Little Boy" bomb dropped on Hiroshima, Japan, ending World War II. Little Boy was a relatively crude device, and its designers were so confident in its construction that Hiroshima was its first full-scale test [Rhodes, 1986 and Rhodes, 1995].

The new nuclear research and production facilities can realistically be hidden in existing industrial parks, where they should escape notice due to the ongoing growth of legitimate industry. Any additional site security required for the nuclear facilities can be hidden from outside observers in a number of practical ways.

C. DATA DEVELOPMENT

To reduce the chance of detection while still achieving sufficient capacity to meet the arsenal goal in a reasonable amount of time, the facilities will be designed to be able to produce five weapons a year. As shown by the calculations in Figures 6 and 7, this production level will require an annual input of 250 kilograms of highly enriched uranium (HEU). Enrichment will thus require an input of about 68 metric tons of yellowcake each year. Because IAEA safeguards for yellowcake cover only imports and exports, covert diversion of this relatively small quantity (5.6 metric tons—one truckload—a month) from the existing production facilities should be easy.

Designing a nuclear weapon and constructing the sophisticated support facilities required to build five weapons a year is a daunting project. Necessary achievements include:

- Covert diversion of 68 metric tons of yellowcake annually;
- production of enrichment plant feed material (uranium hexafluoride, UF_6) from yellowcake;
- uranium enrichment, including the choice of method to employ;
- conversion of highly enriched UF_6 to uranium metal; and
- design and construction of the actual weapons.

Figure 8 displays the tasks included in the case study. We assess the requirements for specialized equipment from the chemical processes shown in Figure 6. There are a total of 194 tasks (nodes) and 583 task partial orders (arcs) in the resulting project network. The proliferator must manage five non-renewable resources: energy, materials,

Technology	Separation Factor	No. Of Stages For 90% HEU	KWh/SWU	KW For 50,000 SWU/Yr
Gaseous diffusion	1.004-1.0045	3,500-4,000	2,500	14,270
Gas centrifuge	1.2-1.5	40-90	100-200	571-1,142
Aerodynamic	1.015-1.030	540-1100	3,600-4,000	20,548-22,831

Table 1. Efficiency of uranium enrichment technologies [after U.S. Congress OTA, 1993, p. 143].
Regardless of the technology used, producing HEU from natural uranium requires multiple equipment stages arranged in a progressive enrichment cascade. The *separation factor* is the ratio of the relative enrichment (U_{235} to U_{238}) of the concentrated product to that of the depleted tails from the output of any one stage of the cascade. The number of stages required to produce HEU assumes the final tails contain less than 0.3% U_{235}. Kilowatt-hours per SWU measures the amount of energy required to produce approximately 5 grams of HEU. The final column is the average power consumption rate of an enrichment cascade producing about 250 kg of HEU a year, enough to construct five fission weapons.

professional labor (e.g., scientists and engineers), skilled labor (e.g., machinists) and unskilled labor. He must choose one and only one enrichment technology to pursue. His choices are gas centrifuges, gaseous diffusion or aerodynamic enrichment. Table 1 lists information used to determine the required number of enrichment machines and energy consumption for each technology. A separative work unit (SWU) is a measure of the effort required to separate U_{235} and U_{238} isotopes during enrichment. Production of 1 kg of HEU from natural uranium requires approximately 200 SWU [Nonproliferation Policy Education Center, 2004].

Our case study employs only mild interdictions, the better to provide a nuanced solution exercise. It is not lost on us that a single Tomahawk missile can render an enrichment centrifuge gallery inoperable, and essentially unoccupiable for millennia.

Feed stock preparation (yellowcake to uranium hexafluoride):

Stainless steel vessel (dissolution of yellowcake in nitric acid):

$$U_3O_8 + 6\ HNO_3 \to 3\,UO_2\left(NO_3\right)_2 + 2\ H_2O + H_2$$

$$842 + 6 \times 63 = 3 \times 394 + 2 \times 18 + 2$$

Stainless steel boiler (boil down of nitrate solution):

$$UO_2\left(NO_3\right)_2 + 6\ H_2O \xrightarrow{BOIL} UO_2\left(NO_3\right)_2 * 6\ H_2O$$

$$394 + 6 \times 18 = 502$$

High-temperature stainless steel boiler (thermal decomposition of nitrate):

$$UO_2\left(NO_3\right)_2 * 6\ H_2O \xrightarrow{300\,C} UO_3 + 6\ H_2O + 2\ NO_x$$

$$502 = 286 + 6 \times 18 + 108$$

Gas-solid reactor vessel (reduction of uranium trioxide):

$$UO_3 + H_2 \xrightarrow{650-800\,C} UO_2 + H_2O \quad \text{or} \quad 3\,UO_3 + 2\ NH_3 \xrightarrow{650-800\,C} 3\,UO_2 + N_2 + 3\ H_2O$$

$$286 + 2 = 270 + 18 \qquad \text{or} \qquad 3 \times 286 + 2 \times 17 = 3 \times 270 + 28 + 3 \times 18$$

Stainless steel reaction vessel:

$$UO_2 + 4\ HF \longrightarrow UF_4 + 2\ H_2O$$

$$270 + 4 \times 20 = 314 + 2 \times 18$$

Ultrahigh-temperature gas-solid reactor vessel (production of uranium hexafluoride):

$$UF_4 + F_2 \xrightarrow{1700-1800\,C} UF_6$$

$$314 + 38 = 352$$

Material conversion (uranium hexafluoride to metal):

Gas-phase reactor with particulate separation:

$$UF_6 + H_2 \xrightarrow{375\,C} UF_4(s) + 2\ HF$$

$$352 + 2 = 314 + 2 \times 20$$

High-temperature metallurgical furnace:

$$UF_4(s) + 2\ Ca \xrightarrow{heat} U(l) + 2\ CaF_2 \quad \text{or} \quad UF_4(s) + 2\ Mg \xrightarrow{heat} U(l) + 2\ MgF_2$$

$$314 + 2 \times 40 = 238 + 2 \times 78 \qquad \text{or} \qquad 314 + 2 \times 24.3 = 238 + 2 \times 62.3$$

Figure 6. Uranium enrichment chemistry [from Harney, 2003].

We study these chemical processes to deduce the facilities and equipment required to produce enriched uranium metal from yellowcake. The numbers beneath the chemical reaction formulas are molecular weights.

1 kg HEU metal requires 1.302 kg $HEUF_4$ and (0.336 kg Ca or 0.204 kg Mg)

1 kg $HEUF_4$ requires 1.122 kg $HEUF_6$ and 0.0064 kg H_2
1.302 kg $HEUF_4$ requires 1.461 kg $HEUF_6$ and 0.0084 kg H_2

1 kg H_2 can be produced from 5.667 kg of NH_3

1 kg $HEUF_6$ requires 232 kg UF_6
1.461 kg $HEUF_6$ requires 339 kg UF_6

1 kg UF_6 requires 0.892 kg UF_4 and 0.108 kg F_2
339 kg UF_6 requires 302 kg UF_4 and 36.6 kg F_2

1 kg UF_4 requires 0.860 kg UO_2 and 0.255 kg HF
302 kg UF_4 requires 260 kg UO_2 and 77.0 kg HF

1 kg UO_2 requires 1.059 kg UO_3 and 0.0074 kg H_2
260 kg UO_2 requires 275 kg UO_3 and 1.924 kg H_2

1 kg UO_3 requires 1.755 kg of hexahydrate
275 kg UO_3 requires 483 kg of hexahydrate

1 kg of hexahydrate requires 0.559 kg yellowcake and 0.251 kg HNO_3
483 kg of hexahydrate requires 270 kg yellowcake and 121 kg HNO_3

Each weapon requires 50 kg 93% HEU metal

5 weapons/yr requires 250 kg HEU metal

250 kg HEU requires:
67,500 kg yellowcake
30,250 kg HNO_3
2.1 kg H_2 (or 11.9 kg NH_3)
19,250 kg HF
9,150 kg F_2
84 kg *Ca* or 51 kg *Mg*

Figure 7. Calculation of annual raw material input requirements [after Harney, 2003].
We assess the quantity of raw materials required to sustain production of five weapons per year. Requirements are computed from the chemical reactions and molecular weights shown in Figure 6.

- **Nuclear Weapon Development**
  - **Divert Commercial Yellowcake**
    - Design yellowcake plant modifications
    - Modify yellowcake plant
    - Divert yellowcake
  - **Produce Enrichment Plant Feed Material**
    - Design fluoridation plant
    - Acquire plant site
    - Prepare plant site
    - **Acquisition Of Plant Components**
      - Stainless steel mixing vessel
      - Distilled water system
      - Nitric acid storage tank
      - Stainless steel boiler
      - Thermal decomposition vessel
      - Drying kiln
      - Gas-solid high-temperature reaction vessel
      - Hydrogen gas (or ammonia) storage tank
      - Stainless steel reaction vessel
      - Hydrogen fluoride storage tank
      - Gas-solid ultra high-temperature reaction vessel
      - Fluorine storage tank
      - Hexafluoride condensing vessel
      - Pumps and piping
    - Assembly and integration
    - Operation
  - **Uranium Enrichment**
    - *Choose enrichment technology*
    - **Gas Centrifuge**
      - Design basic gas centrifuge
      - **Acquire Research Components**
        - Rotor tubes
        - Air bearing systems
        - Motors
        - End caps
        - Centrifuge cases
        - Pumps and piping
      - Assemble research centrifuges
      - Test and evaluation of research centrifuges
      - Design production centrifuges
      - Design enrichment cascade
      - Design enrichment plant
      - Acquire plant site
      - Prepare plant site
      - **Acquire Production Components**
        - Rotor tubes
        - Air bearing systems
        - Motors
        - End caps
        - Centrifuge cases
        - Pumps and piping
      - Assemble production centrifuges
      - Integrate enrichment cascade
    - **Gas Diffusion**
      - Design basic gas diffusion system
      - **Acquire Research Components**
        - Diffusion barriers
        - Heat exchangers
        - Pumps and piping
      - Assemble research devices
      - Test and evaluation of research devices
      - Design production gas diffusion devices
      - Design enrichment cascade
      - Design enrichment plant
      - Acquire plant site
      - Prepare plant site
      - **Acquire Production Components**
        - Diffusion barriers
        - Heat exchangers
        - Pumps and piping
      - Assemble production devices
      - Integrate enrichment cascade
    - **Aerodynamic**
      - Design basic aerodynamic enrichment device
      - **Acquire Research Components**
        - Vortex unit
        - Pumps and piping
      - Assemble research devices
      - Test and evaluation of research devices
      - Design production devices
      - Design enrichment cascade
      - Design enrichment plant
      - Acquire plant site
      - Prepare plant site
      - **Acquire Production Components**
        - Vortex unit
        - Pumps and piping
      - Assemble production devices
      - Integrate enrichment cascade
    - Cascade loading
    - Produce enriched and depleted material
  - **Prepare Uranium Metal**
    - Design metal plant
    - Acquire plant site
    - Prepare plant site
    - **Acquire Plant Components (Enriched Metal)**
      - Gas-phase reactor with particulate collection
      - Hydrogen storage tank
      - Metallurgical furnace
      - Hafnia crucibles
    - **Acquire Plant Components (Depleted Metal)**
      - Gas-phase reactor with particulate collection
      - Hydrogen storage tank
      - Metallurgical furnace

- Hafnia crucibles
- Integrate components
- Produce natural uranium metal
- Produce depleted and enriched uranium metal
- **Prepare Nuclear Explosive Devices**
  - **Design Gun Device Components**
    - Gun
    - Propellant
    - **Critical Core**
      - Fissionable receiver
      - Fissionable projectile
      - Tamper
      - Initiator
    - Safety and arming devices
    - Fuse
  - Design weapon assembly plant
  - Acquire plant site
  - Prepare plant site
  - **Acquire Fabrication Devices**
    - Large-diameter precision lathe
    - Inert-gas environment precision milling machine
    - Metallurgical furnace
    - Hafnium crucibles
    - Inert-gas environment casting system
  - **Acquire Research Device Components (Natural Uranium Prototype)**
    - High-strength steel cylinder
    - Double-base propellant powder
    - Polonium
    - Beryllium powder
    - Detonator and explosive train components
  - **Fabricate Research Device Components (Natural Uranium Prototype)**
    - Gun barrel
    - Breech mechanism
    - Cast uranium components
    - Cast uranium tamper
    - Machine uranium receiver
    - Machine uranium projectile
    - Machine uranium tamper
    - Initiator
    - Propellant charge
    - Detonator and explosive train
  - Assemble research devices (natural uranium prototype)
  - **Acquire Research Device Components (Enriched Uranium Prototype)**
    - High-strength steel cylinder
    - Double-base propellant powder
    - Polonium
    - Beryllium powder
    - Detonator and explosive train components
  - **Fabricate Research Device Components (Enriched Uranium Prototype)**
    - Gun barrel
    - Breech mechanism
    - Cast enriched uranium components
    - Cast depleted uranium tamper
    - Machine enriched uranium receiver
    - Machine enriched uranium projectile
    - Machine depleted uranium tamper
    - Initiator
    - Propellant charge
    - Detonator and explosive train
  - Assemble research devices (enriched uranium prototype)
  - **Test Research Devices (Sub-Critical)**
    - Verify critical mass
    - Verify gun velocity
    - Delivery vehicle compatibility mock-up
  - Test full-scale device
  - Finalize production weapon design
  - **Acquire Weapon Components**
    - High-strength steel cylinder
    - Double-base propellant powder
    - Polonium
    - Beryllium powder
    - Detonator and explosive train components
  - **Fabricate Weapon Components**
    - Gun barrel
    - Breech mechanism
    - Cast enriched uranium components
    - Cast depleted uranium tamper
    - Machine enriched uranium receiver
    - Machine enriched uranium projectile
    - Machine depleted uranium tamper
    - Initiator
    - Propellant charge
    - Detonator and explosive train
    - Fuse
    - Safety and arming device
    - Weapon case and structure
  - Assemble weapon components
  - Production weapons deliveries

Figure 8. Fission weapon development tasks used in our case study [after Harney, 2003].
Summary tasks are indicated in **bold** and the *Choose enrichment technology* decision node is in *italics*.

D. INSIGHTS

We present our enumeration procedure as a tool that provides a quantitative comparison of the myriad options available to an interdictor for senior nonproliferation policy makers. Analysis of an interdictor's options in our case study can take several directions. Ignoring for the moment our two-sided optimization, the most straightforward application answers "what-if" questions manipulating just the proliferator's planning problem. Postulate a delay plan based on current or proposed national policy, inflict it on the project and immediately visualize and evaluate the proliferator's optimal expediting plan in response to that delay plan. Planners can vary input parameters such as the expediting budget, resource availabilities, *et cetera*, over ranges of uncertainty to test the sensitivity of the expediting plan. They might also alter the plan to enhance its consistency with telltale symptoms of its activities, to better assess the true stage of proliferation. This is useful, but gives no information about whether a better delay plan exists.

Using our two-sided enumeration to find the best delay plan for a single combination of input parameters, we can recover the expediting plan resulting from any candidate delay plan that is discovered. This case study reveals many instances of multiple delay plans that all produce the same maximum expedited project length. A set of such delay plans can be presented to policy makers for a more subjective evaluation.

The final mode of study is also the most powerful. We again employ our enumeration procedure to find delay plans, but we now allow input parameters to vary over ranges of uncertainty. The results probe the sensitivity of the choice of delay plan to the inputs as well as the proliferator's response to that delay plan. *The relative frequency with which a particular task appears in the optimal delay plans turns out to be a key indicator of vulnerability and robustness*. If one particular task appears in nearly every optimal delay plan, regardless what the input, that is a clear signal of vulnerability.

Our proliferator has an essentially unlimited supply of yellowcake at his disposal; therefore, our case study does not include the possibility of interdicting the uranium production. When we allow unlimited quantities of the other resources, our proliferator chooses aerodynamic enrichment and completes his project in 260 weeks (five years), given that we do nothing to delay him. If we assume the proliferator has no knowledge

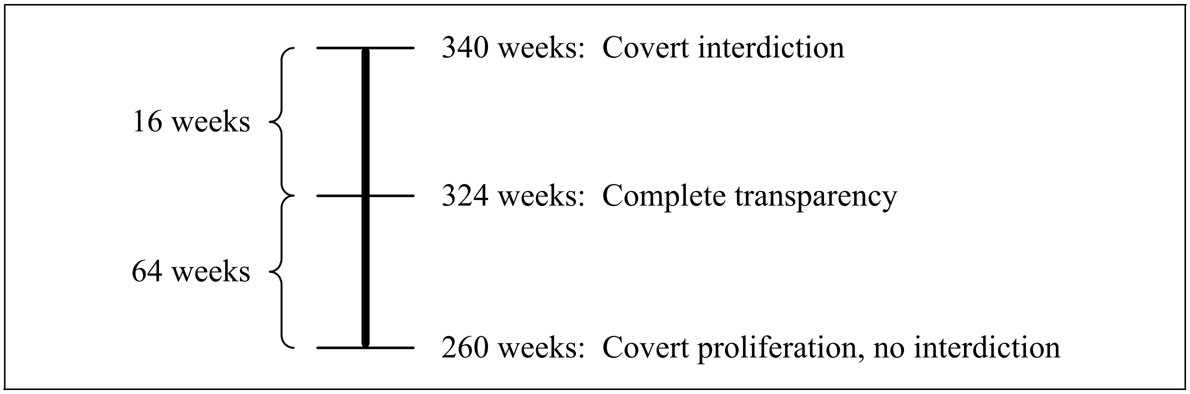

340 weeks: Covert interdiction
16 weeks
324 weeks: Complete transparency
64 weeks
260 weeks: Covert proliferation, no interdiction

Figure 9. Value of information to the proliferator and the interdictor. The proliferator achieves his goal 64 weeks (over a year) sooner by concealing his project from the interdictor. The interdictor extends the project 16 weeks when the interdiction is covert.

of our intention to interdict his project, to maximize the total delay we choose tasks to interdict that are on his critical path. By choosing the two tasks on the critical path with the greatest delay, we can extend the proliferator's project to 340 weeks ($6\frac{1}{2}$ years, a 31% increase), assuming he does not discover or respond to our interdiction. To achieve this delay, we must interdict cascade loading and acquisition of either the vortex unit or pumps and piping for the research enrichment machines.

With an intelligent adversary, our two-sided shows that even if the proliferator is fully aware of our intention to interdict his project and which tasks we will delay, the best he can do is complete the project in 324.4 weeks (about $6\frac{1}{4}$ years). This is a 25% increase over the length of the project when we do nothing. Even though the proliferator knows the project will be interdicted and has free reign to compensate, he is only able to save 15.6 weeks or 4.6% of the project length we achieve by interdicting with complete surprise. The two tasks in the optimal delay plan are cascade loading and acquisition of pumps and piping for the production aerodynamic enrichment machinery. However, because we interdict equipment required for aerodynamic enrichment, the proliferator chooses to save time by using gaseous diffusion instead.

Figure 9 shows the value of secrecy. By maintaining a covert project that cannot be interdicted because no one knows it exists, the proliferator can complete his first nuclear weapon almost 15 months sooner than if he anticipates being interdicted. The interdictor can delay the project 16 weeks longer if he achieves surprise.

Cascade loading appears in all of these delay plans. In every case, with both complete surprise and complete transparency, it is the most vulnerable task in this project. This is useful information to a policy maker or military planner contemplating the obstruction of the proliferator's project.

Figure 10 illustrates the effect on the expedited project length of varying only one input parameter. There is a clear change in the project length when the materials budget is somewhere between $34 and $35 million. Inspection of the resulting expediting plans reveals that the reason for the step is a change in the decision plan. With a small materials budget, the proliferator is forced to choose aerodynamic enrichment, extending the length of the project. Once a sufficient materials budget is provided, he can choose gas-diffusion in order to finish his first weapon more quickly. Figure 11 magnifies the transition in Figure 10. Inflection points like this are easy to discover by systematically searching the domain of resource availability.

Figure 12 shows the effect of varying both the expediting budget and the delaying budget simultaneously. It is clear that no matter what the interdictor does, increasing the expediting budget available to the proliferator reduces the time needed to finish the project. Once the delay budget reaches a certain point, there are no more expensive delaying actions available to the interdictor that cause more delay than the inexpensive one identified with the lower interdiction budget.

Figure 13 shows the effect of varying the expediting budget and the number of interdictions allowed without considering the interdiction budget. Logically, allowing more interdictions admits more delay to the project.

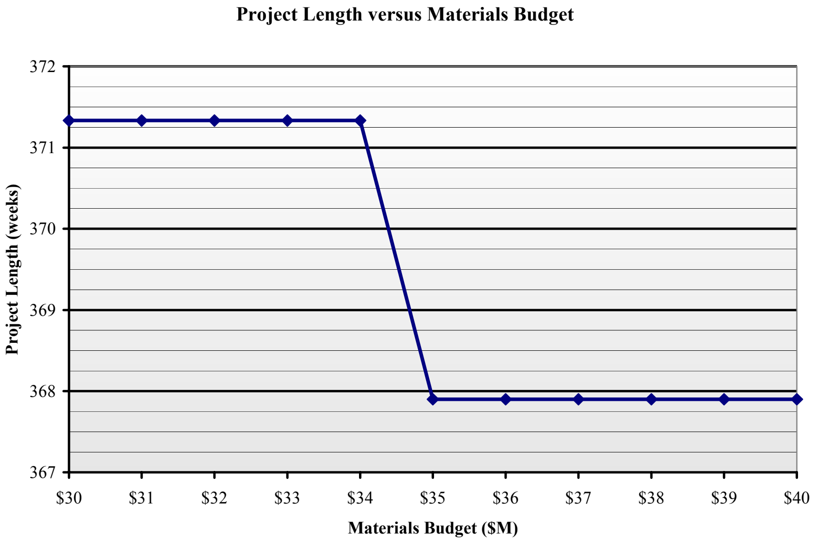


Figure 10. Project length versus materials budget.
The proliferator is allowed a $100M crashing budget, 400 man-months of professional labor and unrestricted amounts of the other resources. The interdictor is allowed two interdictions and $1M delay budget. The step between $34 and $35 million reflects a change in the decision plan. Lower materials budgets force the expeditor to use aerodynamic enrichment while higher budgets encourage use of gas-diffusion to reduce the overall length of the project by 3.4 weeks.

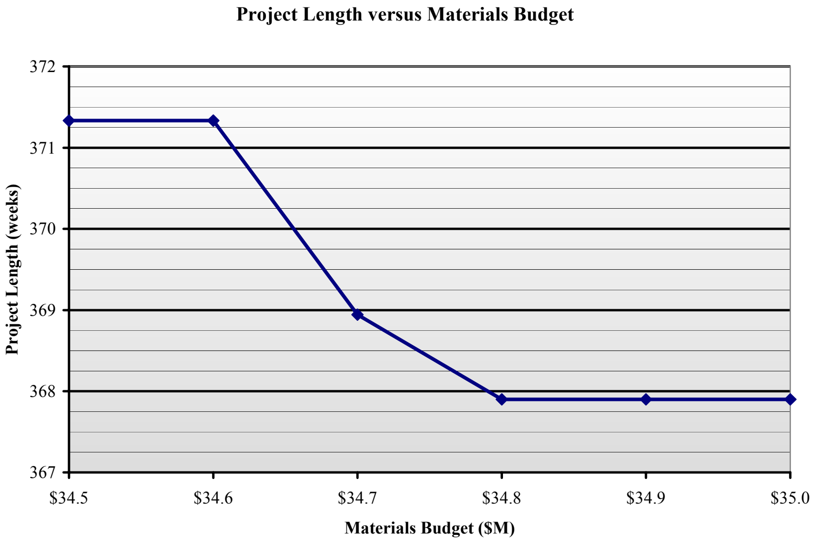


Figure 11. Project length versus materials budget, a closer look.
A closer look at the decision plan transition in Figure 10 shows that the change actually occurs when the materials budget is between $34.6 and $34.7 million.

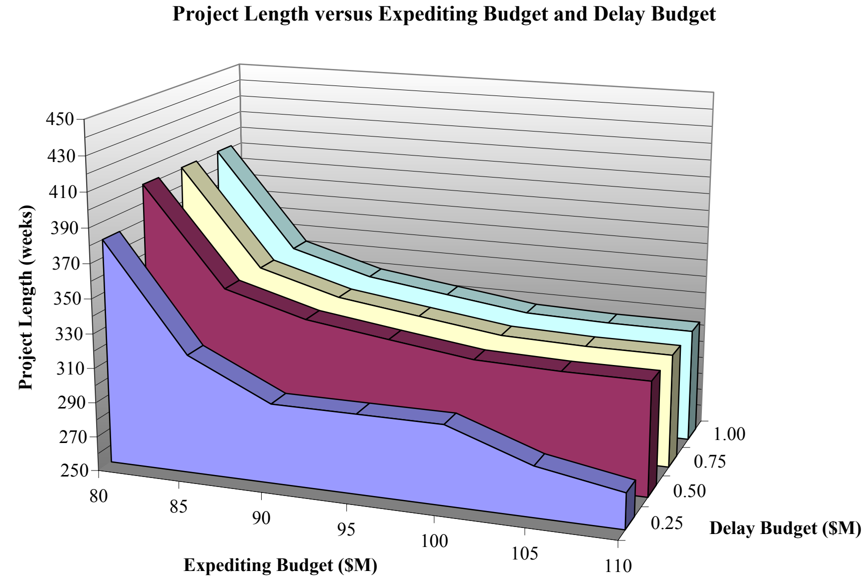


Figure 12. Project length versus budget with one interdiction. Allowing more expensive interdiction actions does not always cause more delay to the project. Once the delay budget reaches $500K, no further improvement in the project delay is achieved. No resource is exhausted here.

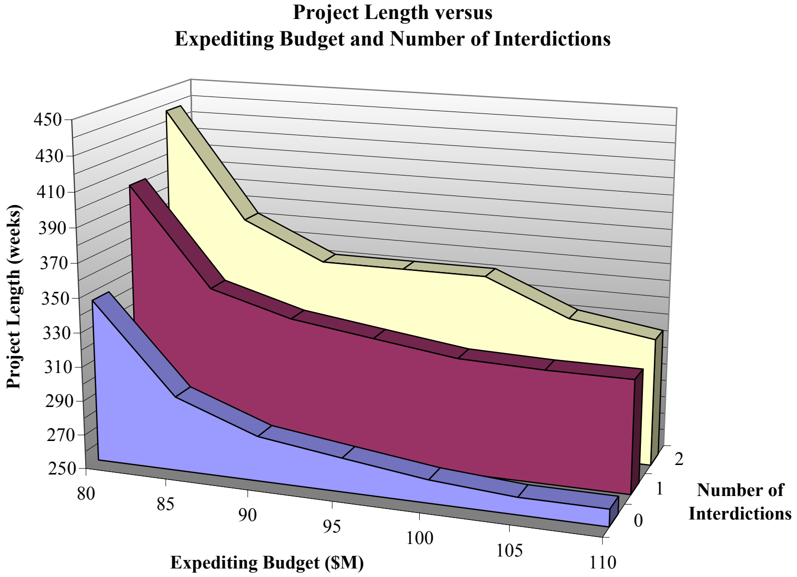


Figure 13. Project length versus expediting budget and number of interdictions. No resource restriction is imposed.

IV. IMPLEMENTATION

We want to show how to solve practical problems, so we have integrated our interdiction procedure with an off-the-shelf project management tool to manage and display the data. The interdiction optimization has been implemented using the General Algebraic Modeling System (GAMS) [Brooke, *et al*, 1998] and XA solver from Sunset Software Technology [2003]. The data is stored and manipulated in a Microsoft Project (MS Project) [Microsoft Corporation, 2003a] database. A custom Visual Basic for Applications (VBA) [Microsoft Corporation, 2003b] user interface hosts the optimization model in GAMS. The VBA code and custom display formatting are saved into an MS Project template file.

By using off-the-shelf software like MS Project, we gain an inexpensive, flexible and easy to use database and display tool. As part of the Microsoft Office [Microsoft Corporation, 2003a] suite of business software, it is widely available and already familiar to a large user base. However, MS Project does not have the sophisticated optimization capabilities that we need. Like all Microsoft Office applications, it permits extensive customization with user-defined graphical user interface elements and VBA code. This feature allows us to easily hide the powerful optimization tools from the user and instead present only a straightforward graphical user interface to control the optimization procedure and display the results.

This choice of software tools does present some challenges, however. The most significant of these is the way MS Project requires a hierarchical structure for the tasks in the project to ease the organization of large projects. MS Project displays each hierarchical task in a Gantt chart [Gantt, 1903] in the style shown in Figure 14. The black bar represents the top-level summary task (the parent) and the subordinate tasks are its children. Summary tasks can be given their own precedence relationship(s) with other tasks in the project. Accommodating them requires introducing dummy precedence arcs into the network to ensure a summary task starts no later than any of its children and to preclude finishing until all of its children finish. (In our model, this requires extra constraints for the binary decision variables.) The summary task must be completed if any of its children are completed and the children must be completed if the summary task is forced to start by other precedence constraints.

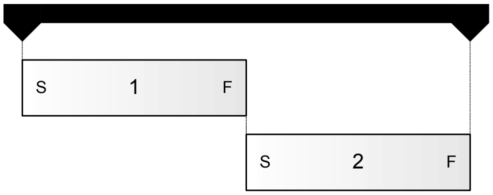
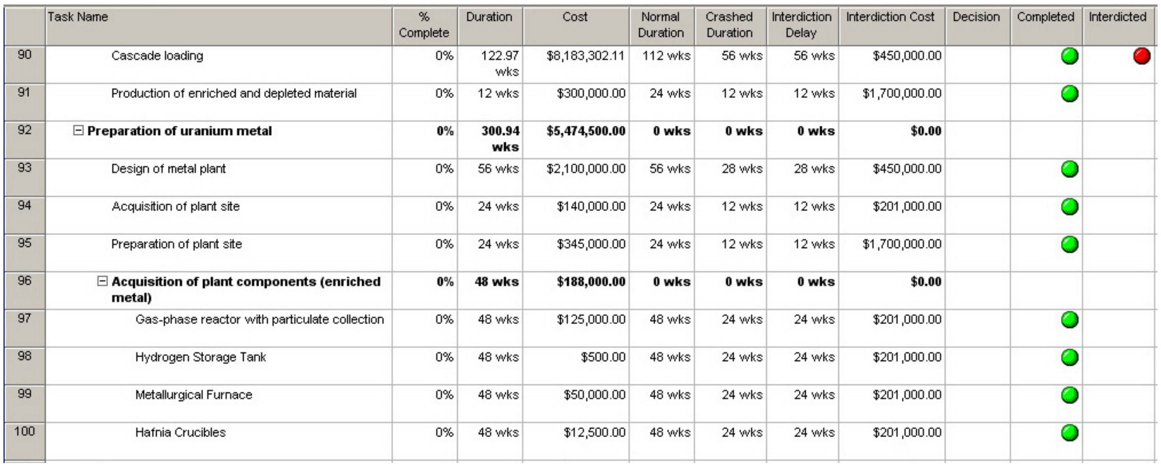


Figure 14. Gantt chart representation of a hierarchical summary task.
The summary task begins as soon as any subordinate task starts and finishes when all subordinates are finished.

Figure 15 illustrates the customized data sheet that is used to enter most of the project data within MS Project. The hierarchical organization of the project tasks is evident in this project view. This sheet also displays the task durations and the decision and delay plans determined by the interdiction procedure. MS Project calculates the cost of each task based on the task duration and the resource requirements and costs in the project database. The data sheet is adjacent to the Gantt chart (Figure 16), enabling quick switching between the tabular data and a graphical representation of the task relationships within the project. Figure 17 shows how tasks in the delay plan are represented on the Gantt chart.

The custom user interface is implemented with VBA code distributed among two user forms (custom dialog boxes) and a code module. The first user form (Figure 18)



	Task Name	% Complete	Duration	Cost	Normal Duration	Crashed Duration	Interdiction Delay	Interdiction Cost	Decision	Completed	Interdicted
90	Cascade loading	0%	122.97 wks	$8,183,302.11	112 wks	56 wks	56 wks	$450,000.00		🟢	🔴
91	Production of enriched and depleted material	0%	12 wks	$300,000.00	24 wks	12 wks	12 wks	$1,700,000.00		🟢	
92	⊟ **Preparation of uranium metal**	**0%**	**300.94 wks**	**$5,474,500.00**	**0 wks**	**0 wks**	**0 wks**	**$0.00**			
93	Design of metal plant	0%	56 wks	$2,100,000.00	56 wks	28 wks	28 wks	$450,000.00		🟢	
94	Acquisition of plant site	0%	24 wks	$140,000.00	24 wks	12 wks	12 wks	$201,000.00		🟢	
95	Preparation of plant site	0%	24 wks	$345,000.00	24 wks	12 wks	12 wks	$1,700,000.00		🟢	
96	⊟ **Acquisition of plant components (enriched metal)**	**0%**	**48 wks**	**$188,000.00**	**0 wks**	**0 wks**	**0 wks**	**$0.00**			
97	Gas-phase reactor with particulate collection	0%	48 wks	$125,000.00	48 wks	24 wks	24 wks	$201,000.00		🟢	
98	Hydrogen Storage Tank	0%	48 wks	$500.00	48 wks	24 wks	24 wks	$201,000.00		🟢	
99	Metallurgical Furnace	0%	48 wks	$50,000.00	48 wks	24 wks	24 wks	$201,000.00		🟢	
100	Hafnia Crucibles	0%	48 wks	$12,500.00	48 wks	24 wks	24 wks	$201,000.00		🟢	

Figure 15. Selected project data from our case study.
The "Decision" column identifies decision nodes with a checkmark. The "Completed" column indicates the decision plan and the "Interdicted" column identifies the delay plan that generated the displayed project schedule.

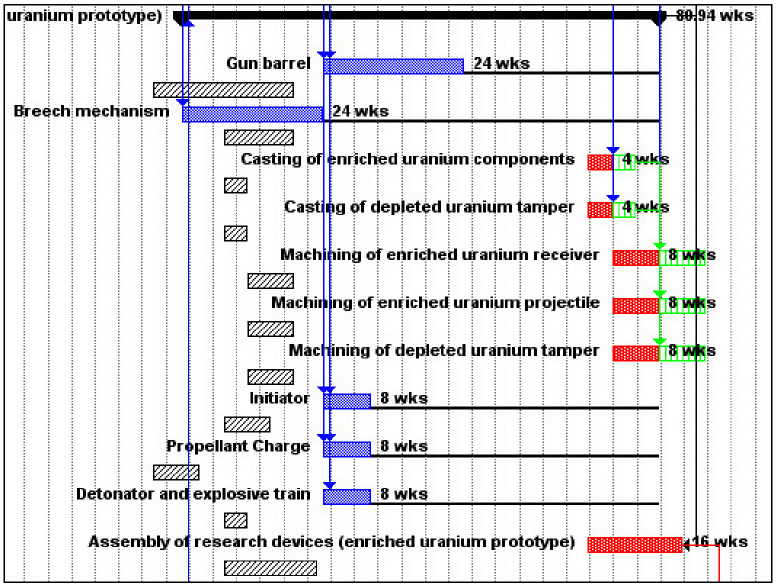


Figure 16. Gantt chart.

Vertical lines delimit four-week time blocks. Each bar with the diagonal slash pattern indicates when a task would be conducted if there is no delay effort. The shaded box above it shows the effect of an interdiction delay elsewhere in the project. The task bars extended with the vertical shading highlight time saved by crashing those tasks. For example, the uranium component casting task has been delayed by 58 weeks and extra resources have been committed to reduce its duration by four weeks.

collects parameters such as the number of iterations desired and the budget, creates data files to be included in the GAMS program and then calls GAMS. Each input parameter, including the individual resource budgets (Figure 19), may be given as a range of values and the GAMS program will automatically solve the problem multiple times and generate a comma-separated values ("CSV") file that can be analyzed using a spreadsheet or mathematical charting and analysis software. The second user form (Figure 20) opens automatically when GAMS terminates or the user presses the "Read Results" button in the first user form. It displays a list containing the length of the project computed in each iteration for each combination of input parameters. Using a quick sort subroutine adapted from VBA-programmer.com [2004], the user can sort the list by the values in any column. Any intermediate result may be selected and read back into the MS Project database for display. The code module provides links to operating system services [Microsoft Corporation, 2004] that permit the optimization software to be called synchronously, allowing the results to be displayed as soon as GAMS terminates.

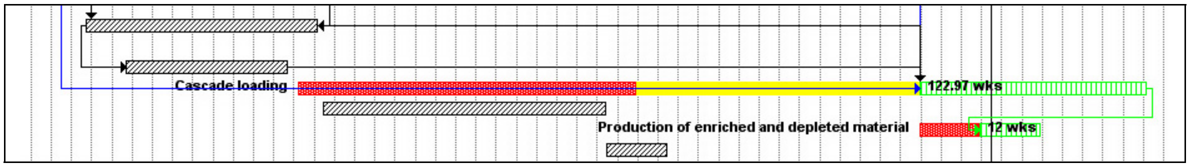
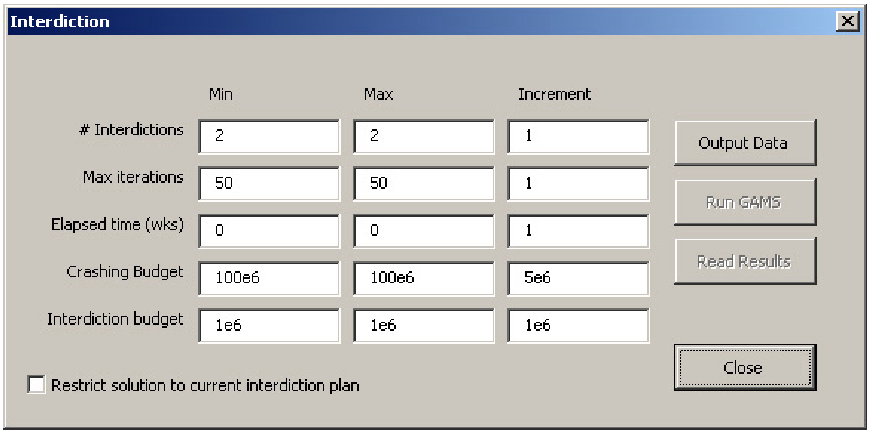


Figure 17. An interdicted task on the Gantt chart.
The lightly shaded section of the task bar between the dark shading and the vertical shading shows the delay imposed on this task by the interdiction effort. The cascade loading task has thus been extended by 56 weeks because of the interdiction.

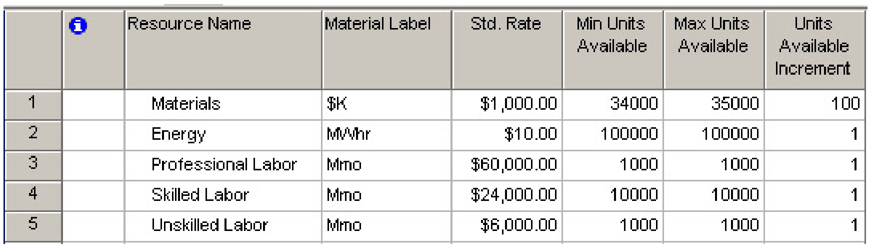
Each instance of this case study requires less than one second to generate, optimize and return to populate results in MS Project. The subproblem uses 193 binary and 385 continuous variables and 1,095 constraints. XA solves this problem to optimality in less than 0.25 second on a Pentium 4 based desktop personal computer. The master problem uses 163 binary variables and two constraints plus one diversity constraint for each prior delay plan. Optimal solution times are similar to the subproblem.



	Min	Max	Increment
# Interdictions	2	2	1
Max iterations	50	50	1
Elapsed time (wks)	0	0	1
Crashing Budget	100e6	100e6	5e6
Interdiction budget	1e6	1e6	1e6

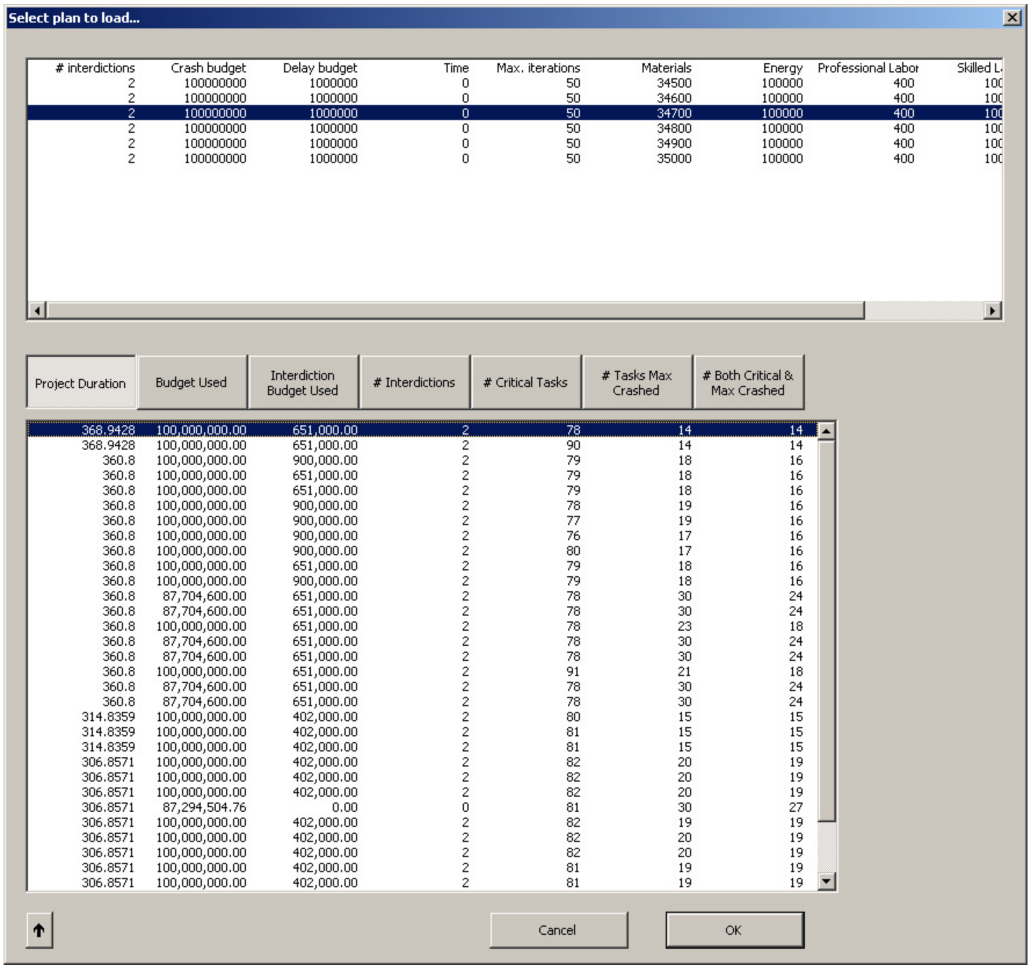
☐ Restrict solution to current interdiction plan

Figure 18. Control panel used to enter parameter ranges for the optimization.
Clicking the "Restrict solution…" check box runs the subproblem to evaluate a particular delay plan rather than finding an optimal plan. This dialog box saves all user input and automatically populates the input fields when opened.



	ℹ	Resource Name	Material Label	Std. Rate	Min Units Available	Max Units Available	Units Available Increment
1		Materials	$K	$1,000.00	34000	35000	100
2		Energy	MWhr	$10.00	100000	100000	1
3		Professional Labor	Mmo	$60,000.00	1000	1000	1
4		Skilled Labor	Mmo	$24,000.00	10000	10000	1
5		Unskilled Labor	Mmo	$6,000.00	1000	1000	1

Figure 19. Resource data.
The "Resource Sheet" view in MS Project is used to enter the ranges for each individual resource budget.



Select plan to load...

# interdictions	Crash budget	Delay budget	Time	Max. iterations	Materials	Energy	Professional Labor	Skilled L
2	100000000	1000000	0	50	34500	100000	400	10
2	100000000	1000000	0	50	34600	100000	400	10
2	100000000	1000000	0	50	34700	100000	400	10
2	100000000	1000000	0	50	34800	100000	400	10
2	100000000	1000000	0	50	34900	100000	400	10
2	100000000	1000000	0	50	35000	100000	400	10

Project Duration	Budget Used	Interdiction Budget Used	# Interdictions	# Critical Tasks	# Tasks Max Crashed	# Both Critical & Max Crashed
368.9428	100,000,000.00	651,000.00	2	78	14	14
368.9428	100,000,000.00	651,000.00	2	90	14	14
360.8	100,000,000.00	900,000.00	2	79	18	16
360.8	100,000,000.00	651,000.00	2	79	18	16
360.8	100,000,000.00	651,000.00	2	79	18	16
360.8	100,000,000.00	900,000.00	2	78	19	16
360.8	100,000,000.00	900,000.00	2	77	19	16
360.8	100,000,000.00	900,000.00	2	76	17	16
360.8	100,000,000.00	900,000.00	2	80	17	16
360.8	100,000,000.00	651,000.00	2	79	18	16
360.8	100,000,000.00	900,000.00	2	79	18	16
360.8	87,704,600.00	651,000.00	2	78	30	24
360.8	87,704,600.00	651,000.00	2	78	30	24
360.8	100,000,000.00	651,000.00	2	78	23	18
360.8	87,704,600.00	651,000.00	2	78	30	24
360.8	87,704,600.00	651,000.00	2	78	30	24
360.8	100,000,000.00	651,000.00	2	91	21	18
360.8	87,704,600.00	651,000.00	2	78	30	24
360.8	87,704,600.00	651,000.00	2	78	30	24
314.8359	100,000,000.00	402,000.00	2	80	15	15
314.8359	100,000,000.00	402,000.00	2	81	15	15
314.8359	100,000,000.00	402,000.00	2	81	15	15
306.8571	100,000,000.00	402,000.00	2	82	20	19
306.8571	100,000,000.00	402,000.00	2	82	20	19
306.8571	100,000,000.00	402,000.00	2	82	20	19
306.8571	87,294,504.76	0.00	0	81	30	27
306.8571	100,000,000.00	402,000.00	2	82	19	19
306.8571	100,000,000.00	402,000.00	2	82	20	19
306.8571	100,000,000.00	402,000.00	2	82	20	19
306.8571	100,000,000.00	402,000.00	2	81	19	19
306.8571	100,000,000.00	402,000.00	2	81	19	19

Figure 20. Dialog box used to load GAMS results into MS Project.
The top list displays every combination of input parameters evaluated. The bottom list displays results of evaluating every candidate delay plan for the parameter combination selected in the top list. This example illustrates that many different delay plans may evaluate to the same overall project length. The header buttons choose which column to sort and the arrow button changes the sort direction.

This case study, along with the MS Project template file and GAMS source code are available from the author or thesis advisor upon substantiated request.

THIS PAGE INTENTIONALLY LEFT BLANK

V. WHAT WE HAVE ACCOMPLISHED

Building on decades of project management research, we present a novel and flexible model that captures many of the issues involved in expediting a sophisticated research and development project. Using this tool as a forecast of proliferation plans, we next seek an optimal set of tasks in the project to interdict in order to inflict the most delay in its completion.

Our prototypic decision support system is implemented using off-the-shelf software. All user interaction and data management are done through software already familiar to millions. Computational experience with our case study shows that good solutions can be found quickly (at less than one second per iteration on a desktop personal computer). "What-if" questions regarding particular delay plans can be answered in a fraction of a second. An analyst can take advantage of this speed to explore a wide range of assumptions and find the tasks that are most critical to the expeditious completion of the proliferator's project.

This low-cost tool can be used *today*. It provides *quantitative* information that can be used to guide policy makers working to stop the proliferation of nuclear, chemical, biological and/or radiological dispersion weapons, or to disrupt more conventional strategic industrial and commercial activities.

Many research opportunities in project network interdiction remain. Modifications to the master problem objective function may yield a faster search for promising candidate delay plans. The design of the enumeration procedure allows a tremendous amount of flexibility with the subproblem and the master problem. We are limited only by what we can express as an integer linear program. Adding additional constraints to the master problem to limit the political, environmental or economic impact of candidate delay plans [e.g., Reed, 1994] is straightforward, as is allowing more than one type of interdiction action for each task. The subproblem could be modified to permit a single interdiction action to delay a group of tasks rather than only one. Our subproblem assumes a linear cost function for crashing, although it could be modified to support nonlinear cost functions [e.g., Vanhoucke, *et al*, 2002] as well.

There is no requirement that the master problem strictly interdict the proliferation subproblem. We can use any integer linear master problem we please to *manipulate* the proliferation optimization to suit our needs. For instance, we might state a master problem that *reschedules* a proliferation effort to maximize its compliance with telltale intelligence about a covert proliferator's actions—evidence of new construction, equipment requisitions, alliances with other proliferators, radiological emissions, power consumption patterns, communications intercepts, *et cetera*. By systematically enumerating alternate proliferation plans, we would hope to identify those that are maximally consistent with what we observe. This immediately yields a tool for evaluating the likely state of a covert weapons program given partial information.

Further refinement of the software developed here could provide valuable functionality. Automating the creation of visual displays such as Figures 11 and 13 would be useful. A tool to visualize changes in the decision plan in response to changing input parameters would be helpful as well. Migrating the optimization model from GAMS to a general-purpose programming language with a directly callable solver package such as the XA callable subroutine library would significantly speed solution times for larger problems, enabling a more thorough exploration of the problem space for the decision maker.

We are suggesting how to organize and manage intelligence data in a cohesive fashion that offers visualization, data base functionality and interoperability with Microsoft Office. The optimization is new, but cleanly hidden from view. An analyst, or even a policy maker, can pose questions, get answers and gain deep insights *without any mathematical background at all*.

LIST OF ABBREVIATIONS AND ACRONYMS

AON	Activity on node
CPM	Critical path method
GAMS	Generalized Algebraic Modeling System
HEU	Highly enriched uranium
IAEA	International Atomic Energy Agency
MS Project	Microsoft Project software
NERAC	Nuclear Energy Research Advisory Committee
NPT	The Treaty on the Non-Proliferation of Nuclear Weapons
OTA	Office of Technology Assessment
PERT	Program evaluation and review technique
SWU	Separative work unit
VBA	Visual Basic for Applications

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Ahuja, R. K., T. L. Magnanti and J. B. Orlin [1993] *Network Flows: Theory, Algorithms, and Applications*, Prentice-Hall, Inc.

Brooke, A., D. Kendrick, A. Meeraus and R. Raman [1998] *GAMS: A User's Guide*, GAMS Development Corporation, Washington, DC, September.

Crowston, W. and G. L. Thompson [1967] "Decision CPM: A Method for Simultaneous Planning, Scheduling, and Control of Projects," *Operations Research*, Vol. 15, No. 3, May-June, pp. 407-426.

Federation of American Scientists [2000] "Osiraq – Iraq Special Weapons Facilities," [http://www.fas.org/nuke/guide/iraq/facility/osiraq.htm], Accessed January 2004.

Gantt, H. L. [1903] "A graphical daily balance in manufacture," *ASME Transactions*, Vol. 24, pp. 1322-1336.

Hamming, R. W. [1986] *Coding and Information Theory*, 2nd ed., Prentice-Hall, p. 189.

Harney, R. C. [2003] "Nuclear Weapon Development Scenario," personal communication, Naval Postgraduate School, Monterey, California, December.

International Business Machines [1968] "Project Management System, Application Description Manual," H20-0210, International Business Machines.

Israeli, E. and R. K. Wood [2002] "Shortest-Path Network Interdiction," *Networks*, Vol. 40(2), pp. 97-111.

Malcolm, D. G., J. H. Roseboom, C. E. Clark and W. Fazar [1959] "Application of a Technique for Research and Development Program Evaluation," *Operations Research*, Vol. 7, No. 5, September-October, pp. 646-669.

Microsoft Corporation [2003a] "Microsoft Office," [http://office.microsoft.com/home/office.aspx?assetid=FX01085795], Accessed February 2004.

Microsoft Corporation [2003b] "Microsoft Visual Basic for Applications Home Page," [http://msdn.microsoft.com/vba/], September 3, Accessed February 2004.

Microsoft Corporation [2004] "How to Determine When a Shelled Process Ends," [http://support.microsoft.com/default.aspx?scid=kb;en-us;178116&Product=vba], Accessed January.

Moder, J. J., C. R. Phillips and E. W. Davis [1983] *Project Management with CPM, PERT and Precedence Diagramming*, 3rd ed., Van Nostrand Reinhold Company Inc.

Nonproliferation Policy Education Center [2004] "Iran: Breaking Out Without Quite Breaking the Rules?" [http://www.npec-web.org/projects/iranswu2.htm], Accessed March 2004.

Nuclear Energy Research Advisory Committee (NERAC) [2001] "Technological Opportunities To Increase the Proliferation Resistance of Global Civilian Nuclear Power Systems (TOPS)," TOPS Task Force Report, January.

Reed, B. K. [1994] *Models for Proliferation Interdiction Response Analysis*, MS Thesis in Operations Research, Naval Postgraduate School, Monterey, California, September.

Rhodes, R. [1986] *The Making of the Atomic Bomb*, Simon & Schuster Inc., p. 561.

Rhodes, R. [1995] *Dark Sun: The Making of the Hydrogen Bomb*, Simon & Schuster Inc., pp. 304-5.

Spears, D. (ed.) [2001] "Technology R&D for Arms Control," *Arms Control and Nonproliferation Technologies*, Lawrence Livermore National Laboratory, Spring, p. D.

Sunset Software Technology [2003] "Sunset Software Technology – Optimization Tools," [http://www.sunsetsoft.com/], August 25, Accessed February 2004.

United Nations [1968] *The Treaty on the Non-Proliferation of Nuclear Weapons*.

United Nations [2004] "Non-Proliferation Treaty," [http://disarmament.un.org:8080/wmd/npt/], Accessed January.

U.S. Congress, Office of Technology Assessment (OTA) [1993] "Technologies Underlying Weapons of Mass Destruction," OTA-BP-ISC-115, U.S. Government Printing Office, December.

Vanhoucke, M., E. Demeulemeester and W. Herroele [2002] "Discrete time/cost tradeoffs in project scheduling with time-switch constraints," *Journal of the Operational Research Society*, Vol. 53, pp. 741–751.

VBA-Programmer.com [2004] "Sorting a Listbox," [http://www.vba-programmer.com/VB_Code/Sorting_Listbox.txt], Accessed January.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Distinguished Professor Gerald G. Brown
   Naval Postgraduate School
   Monterey, California

4. Professor R. Kevin Wood
   Naval Postgraduate School
   Monterey, California

5. Associate Professor Robert C. Harney
   Naval Postgraduate School
   Monterey, California

6. Associate Professor W. Matthew Carlyle
   Naval Postgraduate School
   Monterey, California

7. Anton Rowe
   Naval Postgraduate School
   Monterey, California

8. LT Eric M. Skroch, SC, USN
   Marina, California